AppleWorks Forum

The Monthly Publication of **NAUG:** ***The National AppleWorks Users Group***

Volume VII, No. 10 **Five Dollars**

TABLE OF CONTENTS

Support for AppleWorks and ///EZ Pieces Users

Software Review

TimeOut Grammar Can Improve Your Writing

by Howard Katz

I will never forget my Freshman English professor's critique of my first college paper. After settling down to review the document, I realized that more than half the writing on each page contained critical comments written in barely legible red ink by the professor...and that I had a lot to learn about writing.

Over the years, I think I improved my writing. But I still lamented the lack of a built-in grammar checking program for AppleWorks; a program that would catch my run-on sentences, my mixed tenses and incorrect punctuation, and my excessive use of the passive voice.

Not that grammar checkers were unavailable; Sensible Software's Sensible Grammar program would check my documents, but that meant leaving the AppleWorks environment, running Sensible Grammar, and then returning to AppleWorks. I would use Sensible Grammar occasionally, but going back and forth between the grammar checker and AppleWorks always dampened my enthusiasm for the process.

Now Quality Computers brings us TimeOut Grammar, the latest in the series of TimeOut enhancements for AppleWorks and the first TimeOut product developed since Quality assumed responsibility for these Beagle Bros enhancements. TimeOut Grammar checks your AppleWorks word processor documents for punctuation and style errors without leaving the AppleWorks environment.

Problems with the Language

It is not TimeOut Grammar's fault, but the English language is tortuous to learn and use. As a result, English presents innumerable pitfalls for anyone trying to learn the language...including computers.

To complicate matters, although writers can agree on the essential rules of the language, many great authors consciously or unconsciously violate those rules with spectacular results.

Of course, grammar checkers know nothing about great writing. They work like sophisticated electronic spelling dictionaries and compare your documents to lists of frequently violated rules and incorrect phrases. Thus, grammar checkers might help you prepare a grammatically correct document for a college course or an error-free business letter, but a grammar checker is of little help to a novelist or poet.

Despite the problems associated with the mechanics of the language and the large range of acceptable English writing styles, TimeOut Grammar

Apple-Works Forum

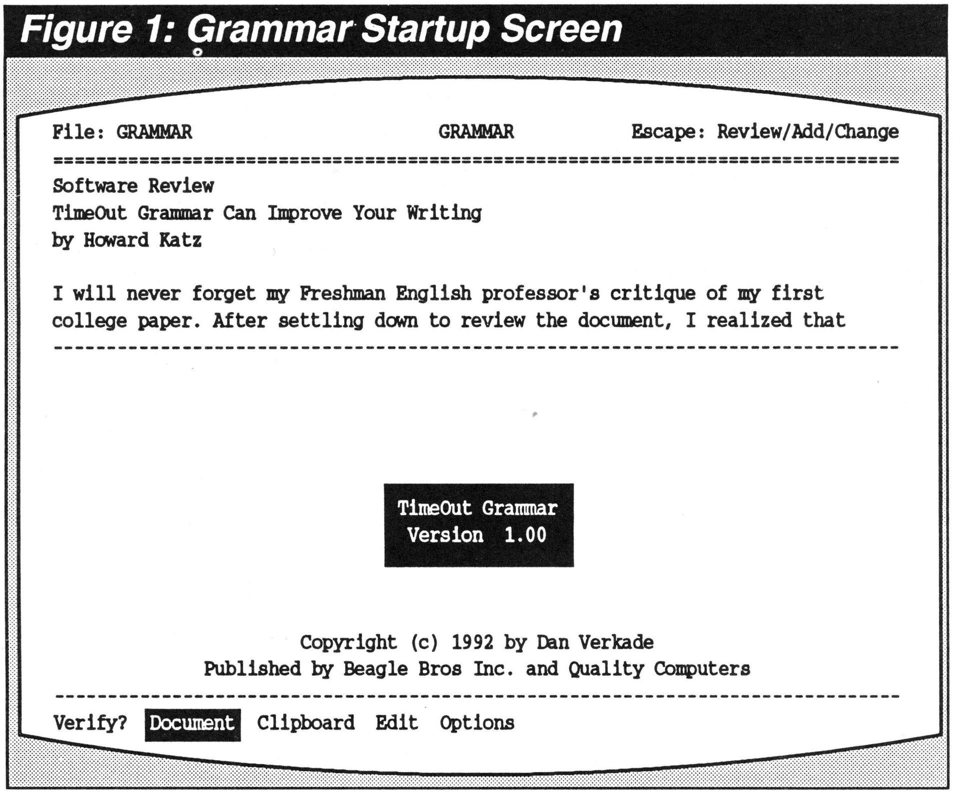
Editor: Cathleen Merritt
Associate Editor: Warren Williams
Page Layout: Nanette Luoma
Publisher: The National AppleWorks Users Group

*©COPYRIGHT 1992, by **NAUG, The National AppleWorks Users Group,** for the exclusive use and enjoyment of its members. Any reprint or reproduction must be approved in writing and in advance by **NAUG.***

*The "**AppleWorks Forum**" (ISSN 0893-4118) is published monthly*
for $31 per year by the National AppleWorks Users Group,
49068 Harvest Dr., Plymouth, MI 48170.
Second Class postage paid at Plymouth, MI, and additional mailing offices.
POSTMASTER: Send address changes to AppleWorks Forum,
NAUG, Box 87453, Canton, MI 48187

The ***National AppleWorks Users Group*** **(NAUG)** is an association that supports AppleWorks users. **NAUG** provides technical support and information about AppleWorks and enhancements to that program. Our primary means of communicating with members is through the monthly newsletter entitled the ***AppleWorks Forum***.

Figure 1: Grammar Startup Screen



```
File: GRAMMAR                        GRAMMAR          Escape: Review/Add/Change
===============================================================================
Software Review
TimeOut Grammar Can Improve Your Writing
by Howard Katz

I will never forget my Freshman English professor's critique of my first
college paper. After settling down to review the document, I realized that
-------------------------------------------------------------------------------

                              TimeOut Grammar
                               Version  1.00

                     Copyright (c) 1992 by Dan Verkade
            Published by Beagle Bros Inc. and Quality Computers
-------------------------------------------------------------------------------
Verify? Document  Clipboard  Edit  Options
```

may prove useful to writers who recognize the limitations of any grammar checking program.

Requirements

TimeOut Grammar runs under AppleWorks 3.0 on any AppleWorks-compatible computer equipped with at least 128K of memory and one 5.25-inch or 3.5-inch disk drive. However, serious users will want a hard disk and at least 200K of unused AppleWorks desktop memory to avoid disk swapping and to speed up the program. The program's design makes it practical for owners of 3.5-inch disk drives, but the delays and disk swapping faced by owners of 128K computers equipped with 5.25-inch disk drives makes the program cumbersome for all but the most occasional use.

Installation

Installing TimeOut Grammar is easy, especially if you have already installed TimeOut on your copy of AppleWorks. All you do is copy the file TO.GRAMMAR and the punctuation and usage subdirectories onto your TimeOut applications disk or subdirectory. The punctuation and usage subdirectories contain common stylistic and punctuation errors in text files that serve as the basis for Grammar's comparative process.

If you have never installed a TimeOut application, the in-elegantly printed but well-written manual gives the necessary instructions. Hard disk users and 3.5-inch disk users without many TimeOut applications will find it easy to copy and configure Grammar for their systems. However, 5.25-inch disk users will need to reconfigure TimeOut for multiple TimeOut application disks; Grammar needs 116K of disk space to store both the application and its lists.

How It Works

TimeOut Grammar works by comparing your document to its list of commonly-used phrases and punctuation errors. Since misspelled words will generate error messages, you must first run your document through a spell checker.

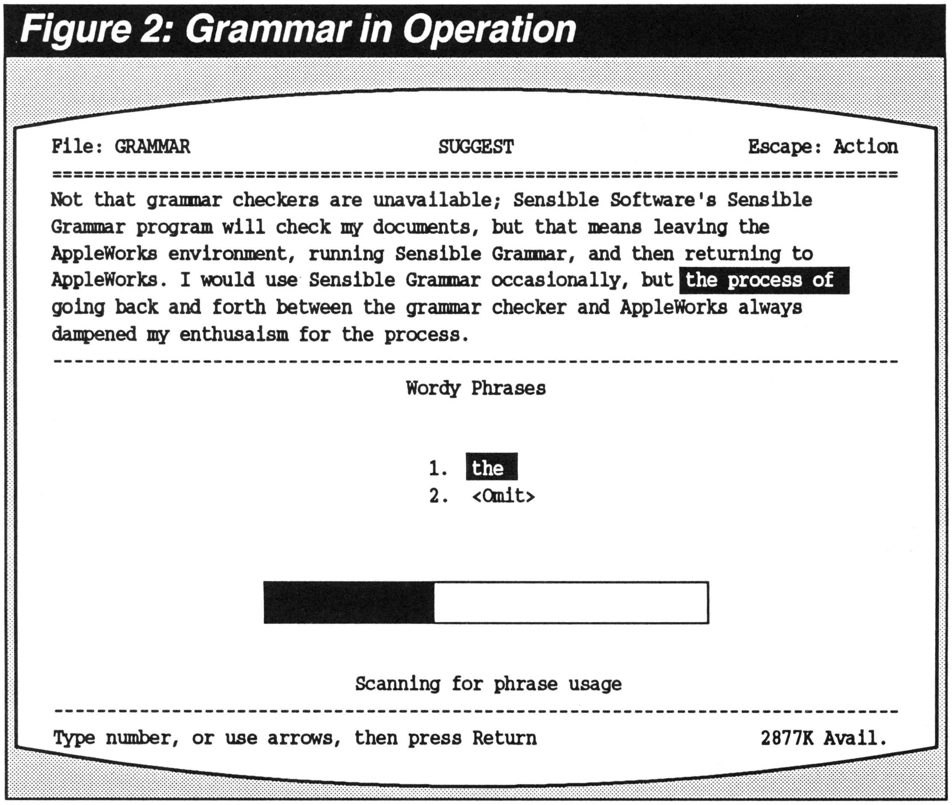
Grammar starts by loading as many of its lists as can fit into the available desktop memory in your system. You will want 200K of available AppleWorks desktop to get the best performance from the program. Otherwise, Grammar will load the lists it can fit into memory, check your document against those lists, load the next set of lists, and repeat this process until it checks the document against all the lists you specify.

Grammar keeps its lists in memory. Although it takes 30 seconds to two minutes to load the Grammar files into memory for the first grammar check, checking additional documents proceeds quickly if you have enough memory in your computer. You can release the grammar lists from memory if you need additional space for your documents.

Operation

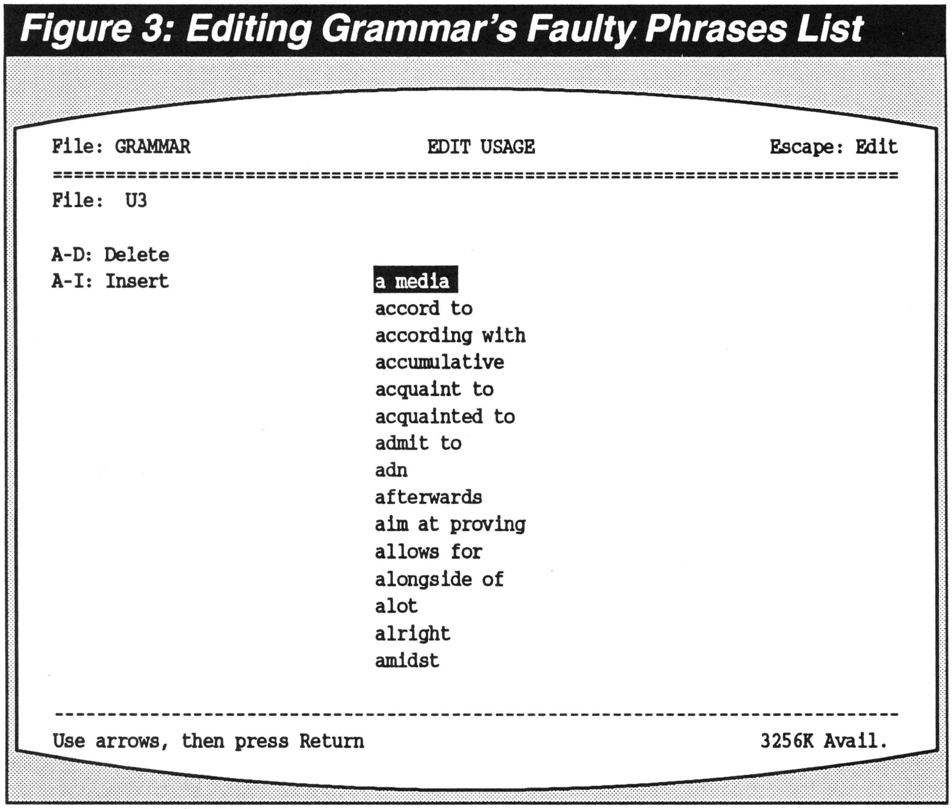
Using Grammar is easy. The menu-driven program starts by asking if you want to check the word processor document on the screen, text stored on the clipboard, or whether you want to edit the phrase and rule lists (see *Figure 1*). The clipboard option lets you check paragraphs and other short segments of

Figure 2: Grammar in Operation



```
File: GRAMMAR                    SUGGEST                    Escape: Action
==========================================================================
Not that grammar checkers are unavailable; Sensible Software's Sensible
Grammar program will check my documents, but that means leaving the
AppleWorks environment, running Sensible Grammar, and then returning to
AppleWorks. I would use Sensible Grammar occasionally, but the process of
going back and forth between the grammar checker and AppleWorks always
dampened my enthusaism for the process.
--------------------------------------------------------------------------
                              Wordy Phrases


                                1. the
                                2. <Omit>




                          Scanning for phrase usage
--------------------------------------------------------------------------
Type number, or use arrows, then press Return               2877K Avail.
```

Figure 3: Editing Grammar's Faulty Phrases List



```
File: GRAMMAR                   EDIT USAGE                    Escape: Edit
==========================================================================
File:  U3

A-D: Delete
A-I: Insert                      a media
                                 accord to
                                 according with
                                 accumulative
                                 acquaint to
                                 acquainted to
                                 admit to
                                 adn
                                 afterwards
                                 aim at proving
                                 allows for
                                 alongside of
                                 alot
                                 alright
                                 amidst

--------------------------------------------------------------------------
Use arrows, then press Return                               3256K Avail.
```

text without checking the complete document. The edit function lets you customize or create your own Grammar phrase lists or change the name of existing categories. The "Options" choice lets you limit the checking to grammar only or punctuation only.

Pressing the Return Key starts the grammar/punctuation-checking process. Grammar searches your document and highlights any suspect phrase. You can then ignore the error, type in a replacement, tell Grammar to display suggested alternatives to the highlighted text, or mark the error and continue checking. Grammar puts brackets around your marked errors, labels each error, and adds a list of your errors to the end of the document.

Grammar's suggestions are particularly helpful and make it easy to improve your writing (see *Figure 2*). Grammar also highlights suspected punctuation errors and lets you ignore, retype, mark, or automatically correct the error.

Grammar also checks and corrects many capitalization errors, including non-capitalized words at the beginning of a sentence and incorrectly capitalized letters within a word. The program recognizes the unique capitalization of the words "AppleWorks", "TimeOut", and some other internally capitalized words. You will find yourself adding other words (including "ReportWriter", "SuperFonts", and "UltraMacros") to the list.

Editing the lists is easy; you choose "Edit" and then indicate what you want to change (see *Figure 3*). Grammar's editing feature lets you customize the program for your own writing style. Whether or not you use this feature depends on how serious you are about your writing; serious writers will want to create separate lists for different types of documents since not everything you write will conform to only one writing style.

Creating those lists is not difficult, but is a time-consuming process. Switching to a new set of lists involves setting up a separate subdirectory and reconfiguring Grammar to look for those lists.

*[Ed: **NAUG** will serve as a clearing house for your customized phrase lists; please send your personalized lists to "Grammar Phrase Lists", **NAUG**, Box 87453, Canton, Michigan 48187.]*

Concerns and Limitations

Unfortunately, Grammar's built-in lists cannot include every exception to the complex rules of the English language. However, a few common rules are curiously absent. For example, Grammar wants me to put double spaces after the period in "Mr." and "Dr." and accepts sentences that start with conjunctions such as "and" and "but". I had to edit the exception lists to correct these oversights.

Since replacing a phrase can cause inconsistencies within a sentence or paragraph, the manual suggests that you run your document through the checking procedure twice. This is true for all grammar checkers; the only difference is that Grammar tells you in the manual while other programs wait for you to discover this problem on your own.

Using Grammar with long documents can be tedious because the program insists on flagging each "error" it encounters, even if you already told the program that the phrase or punctuation is not a concern in your document. That slows down the process as Grammar repeatedly highlights the same phrase or term throughout the document. For example, the phrase "TO.Grammar" generates both punctuation and faulty capitalization error messages. Grammar flagged these errors ten times in an earlier incarnation of this document.

Bugs

By their very nature, grammar checkers are complicated programs, and TimeOut Grammar is no exception. Although generally stable, the program locks up on me occasionally, including every time it encounters a document that contains a Print Date Command.

Documentation

Grammar comes with a 34-page manual that covers installation and program usage. The manual is smaller and does not use the same high quality printing and paper used for the earlier TimeOut documentation. However, the writing style is easy to follow, and the manual contains the information you need to install and use the program. The manual also includes a helpful section with suggestions to improve your writing. The sub-headings in this section include "Passive Voice Is To Be Avoided," "Avoid Cliches Like The Plague," and "Don't Overutilize Big Words," all of which reflect the light-hearted presentations pioneered in the early Beagle manuals and continued by Quality.

Conclusion

Serious writers will find TimeOut Grammar a useful addition to their collection of AppleWorks accessories. Users who spend the time necessary to create different style-type files will find that Grammar can contribute significantly to their writing. Others will use Grammar occasionally to help them tame their bad writing habits and improve their style. Grammar's ability to check your punctuation is a useful tool that you should use as often as you use a spell checker.

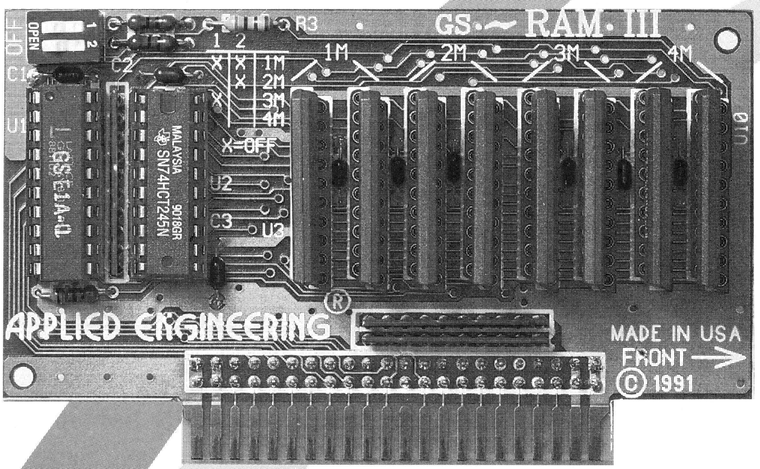
Overall, Grammar does a good job of checking your documents, and the program can help improve your writing. But the complexities of both the program and the English language make the grammar checking process less practical than one might hope.

[Howard Katz is a Medical Underwriter for Travelers Insurance Company in Western Springs, IL.]

*[Quality Computers recently announced the release of version 1.02 of TimeOut Grammar. This version handles abbreviations and the Print Date Command correctly, flags conjunctions at the beginning of sentences, and eliminates the occasional lock-up problem experienced by the reviewer. **NAUG** members can upgrade to version 1.02 by returning their original disk and a self-addressed, stamped return mailer to **NAUG**. We would appreciate, but do not require, a $1 donation to help cover the costs associated with this upgrade program. Our thanks to Quality Computer for providing these update disks.]*

*[TimeOut Grammar costs $45.95 ($79.95 list) plus $3.50 s/h from **NAUG**.]*

It took 250,000 people to bring you GS-RAM III



Does your memory board do all this?

- ✓ Compatible with all standard GS hardware and software
- ✓ Expandable to 6 Megabytes
- ✓ Uses new 1Mb X 4 Chips
- ✓ Accepts 3Mb as valid configuration
- ✓ Low power design
- ✓ Runs cool
- ✓ Compact, efficient design leaves plenty of room for slot 7 cards
- ✓ Fully DMA compatible
- ✓ No jumpers
- ✓ Super Hi-Res diagnostic software included
- ✓ Socketed components for easy upgrade
- ✓ Includes AppleWorks 3.0 print buffer & multi-disk save features
- ✓ Made in USA

In the last 12 years over 250,000 Apple owners have chosen an Applied Engineering memory board to upgrade their system. We've applied our experience and engineering expertise to create the perfect memory board, GS-RAM III. Because of its compact, efficient design, GS-RAM III easily outperforms all other memory boards. It uses the new 1Mb X 4 chips which draw less power than other memory chips or SIMMS. Because the board uses less power, it generates less heat, helping to prevent heat related failures on all of your expansion boards.

Of course GS-RAM III is fully compatible with all standard software and hardware, including DMA devices. It includes our powerful AppleWorks 3.0 expander and diagnostic software. And because we pay attention to detail, GS-RAM III has no jumpers or soldered chips. Socketed components allow for easier upgrades and trouble-shooting.

So if you want more memory for DA's, RAM drives, AppleWorks, AppleWorks GS, HyperCard GS, Platinum Paint, Sonic Blaster or any other program, GS-RAM III is the obvious choice. Join the quarter of a million satisfied Applied Engineering memory board owners. Beam into the next generation of memory boards with GS-RAM III.

To place an order or for more information, see your authorized Æ dealer or call (214) 241-6060 today. Or send check or money order to Applied Engineering. MasterCard, Visa and C.O.D. welcomed. Texas residents add applicable sales tax.

GS-RAM III 1Mb	**$199**
GS-RAM III 2Mb	**$299**
GS-RAM III 3Mb	**$399**
GS-RAM III 4Mb	**$499**
Special GS-RAM III Trade-In ..	**$139**

TRADE-IN

Trade-In *any* manufacturers GS memory board of *any* size and get a GS-RAM III 2Mb for only $139! Call (214) 241-6060 for details.

Æ

APPLIED ENGINEERING®

The Apple enhancement experts.

A Division of Æ Research Corporation

(214) 241-6060
FAX: (214) 484-1365

3210 Beltline
Dallas, TX 75234

Made ★IN THE★ USA

©1992. Æ Research, Inc. All rights reserved. Brand and product names are registered trademarks of their respective holders. All specifications, terms and services are subject to change without notice or recourse.

Now an Even Faster Apple IIGS

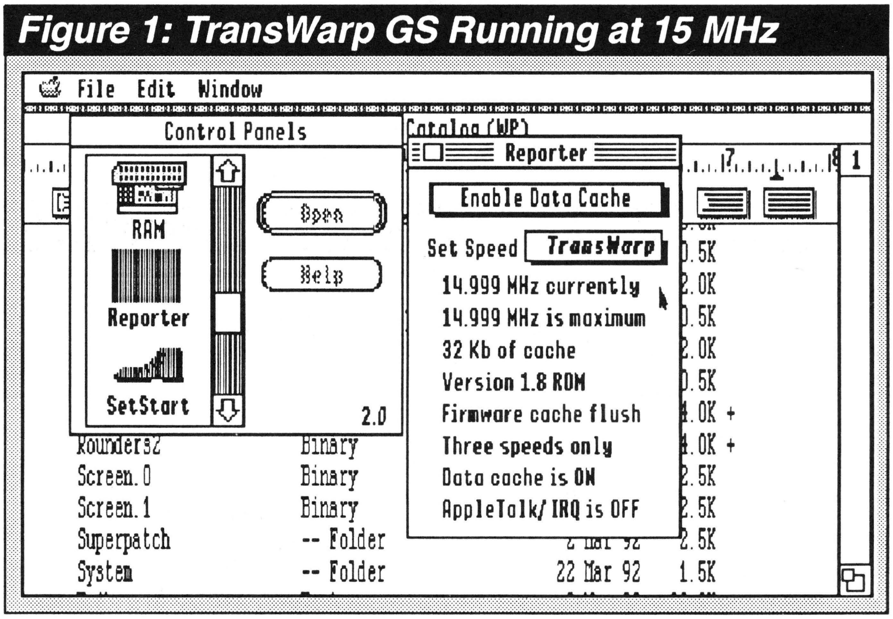
by John Link

*The author assumes that you read his articles about accelerating the TransWarp GS which appeared in the March through May 1991 issues of the **AppleWorks Forum**. The techniques he describes are experimental; you must accept full responsibility for any changes you make to your system.*

The Western Design Center (WDC) recently released a redesigned and faster version of the 65C816 chip that serves as the CPU for the Apple IIGS. WDC rates production-grade parts to run at 10 megahertz (MHz), compared to the 8 MHz maximum speed for the older design. Engineering samples run considerably faster.

The new chips are clearly superior to the original design. The engineering sample I tested runs reliably at 12.5 MHz without adjusting my power supply beyond the normal 5.0 volts. (The engineering sample from the previous design would not run faster than 9 MHz at 5.0 volts.) Furthermore, when I overdrive the new chip, both 8-bit and 16-bit instructions fail simultaneously, suggesting that it may no longer be necessary for designers to set "traps" that slow the execution of certain 16-bit instructions.

Figure 1: TransWarp GS Running at 15 MHz



Set Speed TransWarp
14.999 MHz currently
14.999 MHz is maximum
32 Kb of cache
Version 1.8 ROM
Firmware cache flush
Three speeds only
Data cache is ON
AppleTalk/IRQ is OFF

```
Rounders2      Binary
Screen.0       Binary
Screen.1       Binary
Superpatch     -- Folder
System         -- Folder      22 Mar 92   1.5K
```

Works with "Stock" Accelerators

My TransWarp GS card has the factory 32K cache upgrade installed; that upgrade includes 35-nanosecond (ns) SRAMs. (My original 8K TransWarp GS came with 45ns SRAMs.) I did not need the special 3E Generic Array Logic chip (GAL) or other high speed GALs I described in the May 1991 article until I pushed the chip to a full 15 MHz.

After installing the new chip, my TransWarp GS-equipped system runs reliably at 13.75 MHz with the system voltage set to 5.5 volts. (At 5.0 volts, the system ran at 13.75 MHz for two hours without incident. After that, dial-in calls to my modem failed in 8-bit mode, although I could still dial out successfully, and all 16-bit applications worked reliably.) And, for the first thirty minutes after power up, it ran at 15 MHz (see *Figure 1*)!

I also tested the new 65C816 in a ZipGS with 32K of factory installed cache rated at 45ns. The results were essentially the same; the ZipGS worked reliably at 12.5 MHz at 5.0 volts and at 13.75 MHz at 5.5 volts.

However, the ZipGS would not boot at 15 MHz with the system set to 5.5 volts, nor would it boot at 13.75 MHz with 5.0 volts. (I believe a ZipGS equipped with 35ns SRAMs would boot and might run reliably in those configurations because the ZipGS places less loading on the CPU and thus runs cooler. However, I did not test the ZipGS with the faster SRAMs.)

Figure 2: Oscillator Socket on ZipGS

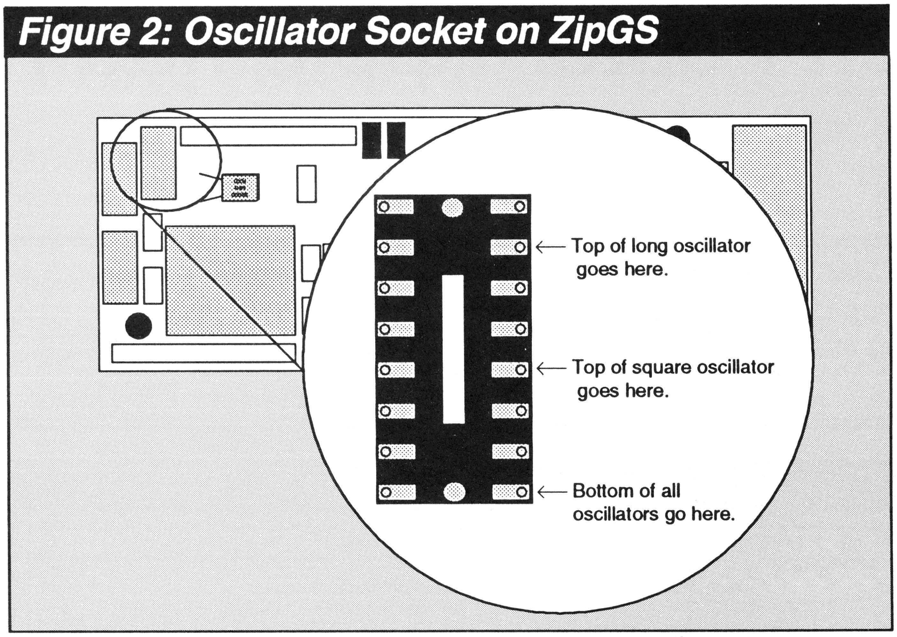
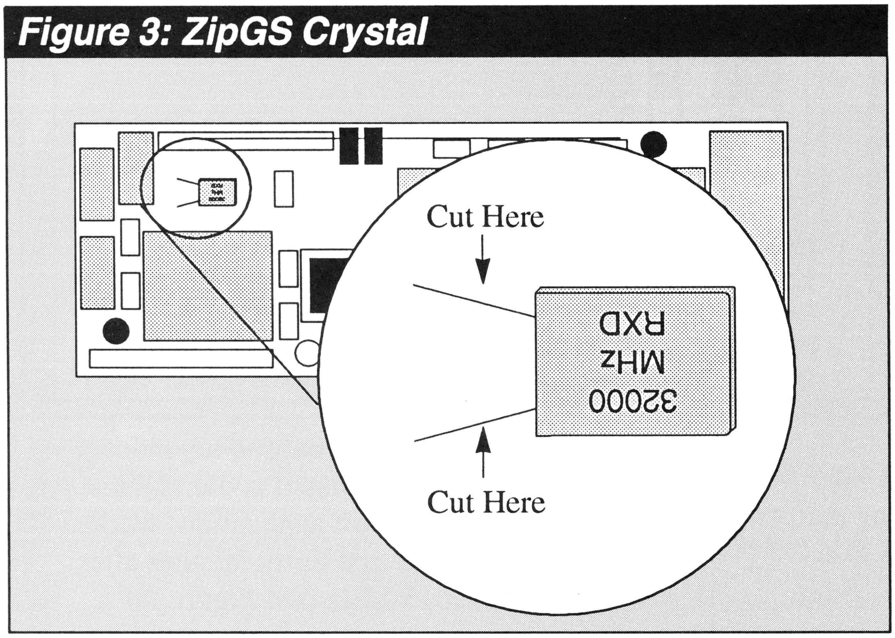

Top of long oscillator goes here.
Top of square oscillator goes here.
Bottom of all oscillators go here.

Figure 3: ZipGS Crystal



Controlling the Voltage

The amount of power available to your system affects its maximum speed. For example, my finding that the new processor operated for two hours at 13.75 MHz at 5.0 volts suggests that users should try the processor at that speed. My tests indicate that some machines provide as much as 5.15 volts from their standard power supply; thus, the new processors might operate reliably at 13.75 MHz in otherwise unmodified systems.

Apple IIGS owners with Applied Engineering power supplies can "tweak" those units to generate more than 5.0 volts. Although Applied no longer manufactures these units, Quality Computers and other mail order dealers stock a similar unit. You can use an insulated screwdriver to change the output from these power supplies without breaking the seal on the outer case. However, I only recommend that procedure for users with the equipment necessary to measure the results and the experience to execute it safely.

As I said in the original articles, you should use a system fan if you install any of these upgrades.

Getting 10 MHz

Most TransWarp GS and ZipGS owners should be able to achieve 10 MHz performance by substituting the new WDC engineering chip for their current part and installing a new 40 MHz oscillator (Digi-Key part #SE1104). I described how to perform these modifications on a TransWarp GS in the April 1991 issue of the ***AppleWorks Forum***.

I tested several sets of 8K 45ns SRAMs, and all worked well at 10 MHz. However, I suggest that you install Applied's 32K cache upgrade for the board; that upgrade includes higher performance 35ns chips which will make it easier to attain reliable operation at these faster speeds.

ZipGS Details

Modifying the ZipGS is similar, except the oscillator socket is at the front of the card (see *Figure 2*).

If your new oscillator comes in a metal case, mount the oscillator with the sharp corner in the upper left hand position. ZipGS owners with metal-cased oscillators do not need the leg extensions that I described for the TransWarp GS; the ZipGS accommodates both the long and square oscillators without modification. You should mount plastic-cased oscillators with the indent facing up toward the top of the ZipGS board.

Figure 4: Scrolling Speeds, AppleWorks GS WP

Speed	TransWarp GS w/ 32K cache*	ZipGS w/ 32K cache*
15.00 MHz	39.0	N/A
13.75 MHz	40.9	47.7
12.50 MHz	45.0	51.8
10.00 MHz	48.2	56.2
7.00 MHz	61.2	70.1

* *Time in seconds; lower numbers are better.*

Make certain that you put the bottom legs of the oscillator in the bottom-most holes in the socket. No matter which oscillator you install, there will be empty holes in the upper part of the socket, as shown in *Figure 2*.

Some early production models of the ZipGS came with a soldered crystal and an empty oscillator socket. Owners of those units must clip the crystal's leads to detach it from the board (see *Figure 3*).

The ZipGS holds its processor firmly, and removing the chip takes some patience. You can use the basic prying technique that I described in my April 1991 article about the TransWarp GS, but you must be patient and careful. I managed to remove the processor with a bent paper clip. However, the best solution is to use a 44-pin PLCC chip puller that you can get from an electronics supplier.

Getting 12.5 MHz

I achieved 12.5 MHz performance from both the TransWarp GS and the ZipGS simply by substituting a 50 MHz oscillator (Digi-Key part # SE1108) for the 40 MHz unit; I did not need to increase the system voltage beyond the standard 5.0 volts. TransWarp GS owners with 45ns SRAMs must install the 32K cache upgrade to get the 35ns SRAMs that seem necessary for this level of performance.

The ZipGS design lets it use fewer and slower SRAMs for a given cache size and operating speed. Thus, my Zip accelerator worked well at 12.5 MHz with its original 45ns parts. However, the Zip is not as efficient as the TransWarp GS, so the Zip accelerator performs somewhat slower than the corresponding TransWarp accelerator. (See *Figure 4* which lists the time it takes to scroll through a 14-page AppleWorks GS word processor document with different processors.)

Getting 13.75 MHz

Both the TransWarp GS and ZipGS accelerators performed reliably at 13.75 MHz with the new chip. Whether or not you need a new power supply to get this performance depends on the output of your current unit. I suggest that you check your system by installing a 55 MHz oscillator (Digi-Key part # SE1109) and the new engineering grade 65C816 and see what happens. If your system fails, you can try to obtain 13.75 MHz performance by installing the AE-type power supply and having a competent technician raise the output to 5.5 volts. TransWarp GS owners will need the faster SRAMs included in the 32K cache upgrade to achieve reliable 13.75 MHz performance. ZipGS owners do not have to change any other chips on the board if they work as well as mine do. Otherwise, you can buy 35ns SRAMs for about $15 each, but you must be careful to get the correct size. Some versions of the ZipGS use wide body chips, others use the narrow body versions, and some ZipGS accelerators use both sizes.

Getting 15 MHz

I got my TransWarp GS to run briefly at 15 MHz by replacing the standard GALs with the high speed GALs described in my May 1991 article and by installing a 60 MHz oscillator (Digi-Key part # SE1110). Even then, I experienced problems after 30 minutes at 5.5 volts, which is the upper limit of the voltage that I will run in my system. I did not substitute faster SRAMs because I suspected the processor itself was reaching the edge of its potential. In addition, the TransWarp GS puts a significant load on the chip, which generates additional heat that degrades the upper performance limit.

The ZipGS puts less of a load on the chip, and with faster SRAMs it might perform reliably at 15 MHz. However, I could not get it to work at all with the factory supplied 45ns SRAMs. Since my scrolling tests revealed that a TransWarp GS running at a given speed outperforms a ZipGS running 2 MHz faster, the possibility of attaining 1.25 MHz greater speed did not seem worth the effort.

Qualifiers

While a system fan is a necessity for any high speed system, the GALs and other special parts I described in my earlier articles do not seem necessary for users who are satisfied with 13.75 MHz performance from the new processor. However, you will probably need more than the standard 5.0 volts to get 13.75 MHz operation, whether that comes from a standard power supply that is on the high side, or from a special unit. You must also remember that each engineering chip, accelerator card, and Apple IIGS computer is unique. Your results will probably differ from mine, especially if you try to get the absolute maximum speed that your chip, accelerator card, and system voltage can support.

For these reasons, I limited my testing to the standard 5.0 volts and the maximum 5.5 volts that I use in my own computer. My results at these two voltages can serve as guidelines for anyone who wants to modify their system. However, they are not absolute predictors of the performance you will get after making these changes.

Real World Effects

Although I do not regularly recalculate spreadsheets or sort data bases, I scroll around word processor documents all the time, and I do not like to wait when I perform this task. So I scrolled through a 14-page AppleWorks GS document to test the real world differences between each level of processor speed. *Figure 4* summarizes the results I attained with each accelerator. AppleWorks Classic users will find similar speed improvements in data base searching and sorting, and in spreadsheet recalculations. Video performance in AppleWorks Classic is phenomenal. At 15 MHz I scrolled through every line in a 4800 line document in slightly more than 6 seconds.

Conclusion

As you can see from *Figure 4*, doubling the speed of an accelerator does not halve the time it takes to perform an operation. For example, going from 7 MHz to 15 MHz on a 32K TransWarp GS (a 214 percent increase in speed) decreases scrolling time by 64 percent. Speeding up the ZipGS has about the same percentage effect, although it starts from a slower baseline.

Increasing cache size to 32K remains the most cost effective modification you can make to your accelerator. However, the new WDC chips now make it easy and inexpensive to achieve speeds greater than 10 MHz on your Apple IIGS. Because the cache and processor upgrades support each other, the performance gains you get by making both modifications will exceed the sum of each taken separately. Besides, it is a lot more fun to say you have a 13.75 MHz IIGS (why not round that off to 14 MHz?) than to say that your computer has 32K of accelerator cache. ▩

[John Link is an AppleWorks consultant and the developer of SuperPatch and LockOut. Mr. Link has written more than 30 articles about Apple II, Macintosh, and NeXT computers.]

*[NAUG members can buy the new engineering-grade chips for $71.25 plus s/h directly from the manufacturer. List price for these chips is $95. Contact WDC's new electronic bulletin board at (215) 646-7247 for more information. **NAUG** members without modems should contact WDC at (215) 646-3025 (voice) or (215) 646-6083 (fax) before ordering. Do not send checks or purchase orders without first contacting the company.]*

[Wester Design Center, 2166 E. Brown Road, Mesa, AZ 85213; (602) 962-4545; Fax: (602) 835-6442.]

[Digi-Key Corporation, 701 Brooks Avenue South, P. O. Box 677, Thief River Falls, Minnesota 56701; (800) 344-4539; Fax: (218) 681-3380.]

*[NAUG members experimenting with different processors on TransWarp GS cards should get John Link's TWGS Reporter. TWGS Reporter is a GS/OS CDEV that reports the characteristics of your TransWarp GS accelerator (see Figure 1). The TWGS Reporter Disk costs $6 plus $2 s/h from **NAUG**.]*

UltraMacros Primer

Converting Your Macros for Ultra 4

by Mark Munz

*This is the first in a series of articles that will help UltraMacros users learn the new features of Ultra 4. The author assumes that you can create macros. **NAUG** members not familiar with UltraMacros should read Mark Munz's book, **The UltraMacros Primer**, which costs $17.95 plus $3.50 s/h from **NAUG**.*

Ultra 4 adds important features to UltraMacros and changes some basic elements of the UltraMacros programming language. *[Ed: A brief description of these changes appeared in the review of Ultra 4 published in last month's issue of the **AppleWorks Forum**.]* These changes make many macros written for UltraMacros 3.x incompatible with the new Ultra 4 program.

To ease the transition to the new version of UltraMacros, Randy Brandt designed the program so that UltraMacros 3 and Ultra 4 can co-exist on the same AppleWorks 3.0 disk. Although you can only run one of these enhancements during each AppleWorks session, you can always quit AppleWorks and switch to the other version of UltraMacros.

The ability to switch between macro enhancement programs lets you continue to use your current macros and commercial macro-based applications such as AlphaCheck and EuroWorks while developers create Ultra 4-compatible versions of their programs. However, you will want to convert your macros so they run under Ultra 4.

Although Ultra 4 offers new commands and features, many of your current macros will prove compatible with the program. This article describes how to convert your old macros into the new Ultra 4 format.

This is a four-step process:

1. Document your macros with comments.
2. Rename all <ba-ctrl-> macros.
3. Replace all commands not supported by Ultra 4.
4. Re-compile and test your macros and task files.

The Files Used by the Two Programs

Here are the files used by the two versions of UltraMacros:

UltraMacros 3.1	Ultra 4
ULTRA.SYSTEM	UM4.0.SYSTEM
TO.COMPILER	TO.UM.COMPILER
TO.MACRO.OPT	TO.UM.OPTIONS
TO.DEBUG	I.DEBUG (an init)

Getting Started

You make these changes to the "source files" for your macros. The source files are AppleWorks word processor documents that contain a readable version of the macros. Most users keep copies of their macros in AppleWorks word processor documents. Users who do not keep the original source files can use the UltraMacros compiler to re-create those files.

Follow these steps to create word processor files that contain your macros:

1. Launch ULTRA.SYSTEM to start the 3.x version of UltraMacros.
2. Create a new AppleWorks word processor document called Task.xxxxxxxxxx where you replace the "x's" with the name of the task file.
3. Press Apple-Escape to display the TimeOut Menu and launch the Macro Compiler.
4. Select choice #2, "Display current macro set". The compiler will display the current macro set in the word processor file on your desktop.
5. Issue an Apple-S command to save that file.

Figure 1: Ultra 4 Replacements for <inc> and <dec>

```
<sa-I>:<all:w=1:ba-*>!     // <inc> command

<sa-D>:<all:w=Ø:ba-*>!     // <dec> command

<ba-*>:<all:
  z=peek #socursor :       // remember the cursor type
  insert : oa-e :          // force to the overstrike cursor
  $Ø="" : read : left :    // read in the character under the cursor
  y=asc $Ø :               // get the ascii value
  if w=1 then : y=y+1 :    // if w=1, increment the character value
  else y=y-1 : endif :     // otherwise decrement the character value
  if y>31 and y<127 :      // verify that it is still a legal character
    print chr$ y :         // print the new character
  else : right : endif :   // otherwise skip it since it's not a real character
  if z=Ø : then : oa-e : endif: // revert to the insert cursor if it's appropriate
>!
```

Figure 2: Ultra 4 Replacements for "&" Commands

UltraMacros 3	Ultra 4
& "space"	.spacebar
& "y/n" : & "Do This"	.menubar "Do This", "No, Yes":
& "index"	.cls 1 : oa-Q :
& "path" : & "/newpath"	.setdisk "/newpath"

6. Go to the TimeOut Menu and launch "Macro Options".
7. Press the Return Key to choose "Launch a new task" and select a new task file from the list that appears on your screen.
8. Repeat steps #2 - 7 until you convert each task file into an AppleWorks word processor document.

Adding Comments

Start by adding comments that document your macros. Ultra 4 treats all text that follows two slashes as a comment. For example:

```
x:<all:oa-Q:esc: // You can put anything in a comment!
rtn:rtn>!
```

Go through your macros and insert Returns to separate each macro onto multiple lines. Then add comments to document your macros.

Macro Names

Since Ultra 4 does not support Both-Apple-Control <ba-ctrl-> macros, you should rename your <ba-ctrl-> macros as <sa->, <ba->, or <sa-ctrl-> macros.

Edit the Macros

Next, edit your macros to eliminate the following commands not supported by Ultra 4:

<inc> and <dec>: These commands increment and decrement the character under the cursor. Replace these commands with either the <sa-I> or <sa-D> macro and the <ba-*> macro listed in *Figure 1*. Both <sa-I> and <sa-D> call <ba-*>, which duplicates the function of <inc> or <dec> depending on the value of the variable W. You can assign any names to these macros.

<elseoff>: Replace this command with <endif>.

<&>: Ultra 4 replaces the ampersand command with dot commands which can themselves accept parameters. The Ultra 4 equivalents for these commands appear in *Figure 2*. I will describe the syntax of the various dot commands in a later article in this series.

<ifkey>: Ultra 4 does not support <ifkey>, which requires the user to press a designated key. *Figure 3* demonstrates how to use <peek $CØØØ> in place of <ifkey> in your Ultra 4 macros.

<getvar> and <putvar>: Ultra 4 does not store numeric and string variables in sets and does not offer <getvar> and <putvar> commands. See the section entitled "Managing Variables" below for information about how to convert the <getvar> and <putvar> commands for Ultra 4.

<cls>: Replaced by the dot command ".cls *value*". Replace <cls> with <.cls 1>. The "1" signifies clearing the normal area. I will describe the syntax of the <.cls> and other dot commands in a later article in this series.

<id#>: Replaced by the dot command ".id". Change <x= id#> to <x = .id>.

Figure 3: Converting UltraMacros 3's <ifkey>

```
UltraMacros 3                                            Ultra 4

x:<all:ifkey rtn then print "Return was pressed">! ——► x:<all: z=peek $CØØØ: if z= 13 then print "Return was pressed">!

y:<all:ifkey>A<then print "Letter A was pressed">! ——► y:<all: z=peek $CØØØ: if z=#'A' then print "Letter A was pressed">!
```

Figure 4: Converting UltraMacros 3's <menu> Command

These macros create four-item menus starting at location x=5, y=5 with double spacing and with item #2 as the first highlighted item.

```
UltraMacros 3                                        Ultra 4

x:<all:                                              x:<all:
  poke #menufirst,2: { make item #2 first highlighted}  $1 = "Item 1":
  poke #menuinc,2 :  { double-spacing }                 $2 = "Item 2":
  poke #menuhor,5 :  { x=5 }                            $3 = "Item 3":
  z=5 :              { y=5 }                            $4 = "Item 4":
  menu "Item 1":                                        .makemenu 5,5,1,4,2,2:
  menu "Item 2":                                        // The syntax for .makemenu follows:
  menu "Item 3":                                        // .makemenu x, y, start string, # of items, spacing, first item
  menu "Item 4":                                     >!
>!
```

Figure 5: Parameters for the <clear> Command

```
Syntax: <clear #>
 #
0-9      clear numeric array with subscript #. Thus, <clear 2> clears variables A(2) through Z(2).
50       clear all numeric variables.
100-190  clear ten strings starting at the value you enter minus 100. Thus, <clear 120> clears strings $20-$29.
200      clear all string variables.
255      clear all numeric and string variables.
```

<findpo>: Replaced by the dot command ".findpo". Change <findpo> to <.findpo>.

<menu>: Replaced by a new set of dot commands that create menus. Follow the syntax of the example in *Figure 4.*

<call>: The <call> and <jsr> commands perform the same functions in UltraMacros 3. Ultra 4 redefines <call> so it calls a macro as a subroutine from another task file. See the sidebar entitled "Shortening Your Macros" for more information about Ultra 4's <call> command.

Replace all your UltraMacros 3 references to <call> with <jsr>. For example, <call $300> becomes <jsr $300>.

<clear>: In UltraMacros 3, this command sets all numeric variables to zero and all string variables to "" (no definition). In Ultra 4, <clear> takes a value parameter that lets you clear a designated portion of the variables. The chart in *Figure 5* lists the values you can use to clear portions of either numeric or string variables.

Change all UltraMacros 3 references from <clear> to <clear 255> in Ultra 4.

<find>: Ultra 4 includes significant changes to the <find> command; I will describe those changes in a later article in this series.

For now, note that <find> no longer zooms out first in the word processor but can find Return "blots" without zooming.

Change all UltraMacros 3 references to <find> as follows:

When	UltraMacros 3	Ultra 4
At a menu:	<find>	<z=2ØØ : find>
In the WP:	<find>	<.zoomin : find>

(The <.zoomin> Dot Command does the reverse of <zoom>. <.zoomin> zooms in to display the printer options in word processor documents, formulas in the spreadsheet module, and single record layout in the data base module. I will describe <.zoomin> in a later article in this series.)

<uc> and <lc>: These two commands no longer change the cursor to an overstrike. That means you no longer have to preserve the insert or overstrike cursor when using <uc> and <lc>. However, you do not have to change your macros; older macros that use these two commands will work properly in Ultra 4.

<onerr>: In UltraMacros 3, <onerr stop> stopped the current macro and returned control to the calling macro. In Ultra 4, the <onerr stop> option stops all macros that are running. To stop a single macro, replace <onerr stop> with <onerr endmacro>.

Ultra 4 also offers <onerr exit> which exits a begin/rpt loop when it encounters an error.

<oa-ctrl-x>: Ultra 4 does not support <oa-ctrl-x> as an equivalent for the <clear> command. Instead, it uses the <debug> command that I will describe in a future article.

Replace all <oa-ctrl-x> commands with <clear 255>.

Managing Variables

Your Ultra 4 macros can include up to ten times as many variables as your earlier macros. However, Ultra 4 handles numeric and string variables differently than earlier versions of UltraMacros. Converting your variables will be one of the more challenging tasks you face when transforming your macros to run under Ultra 4.

String Variables: Ultra 4 supports up to 100 string variables ($0-$99). You can access string variables just as you access variables $0 through $9. For example:

```
 x:<all:  $1 = "This is string #1":
          $10= "This is string #10":
          $23= "This is string #23":
          $99= "This is string #99":
print $99>!
```

Numeric Variables: Ultra 4 supports arrays of numeric variables such as A(Ø), A(9), and K(5), each of which can store a different number. Ultra 4 will treat your current references to variables A to Z as references to variables A(Ø) to Z(Ø).

Since Ultra 4 does not use sets of variables, the program no longer supports the <getvar> and <put-

Shortening Your Macros

Ultra 4 offers a smaller macro table (3,984 bytes) than UltraMacros 3.1 (4,001 bytes). Ultra 4 also uses more macro space for begin/rpt loops and if-then-else commands than earlier versions of UltraMacros. If your macros use close to the maximum number of bytes allowed in UltraMacros 3.1, you may have to shorten them for Ultra 4.

The quickest way to shorten your macro table is to remove the macros you do not use and re-compile.

If you cannot discard any macros, you can put some macros in a separate task file and use the Ultra 4 <call> command to call those macros. Use the syntax

```
    <call MACRO in "taskfile">
```

where you replace "MACRO" with the keystrokes that call the macro and "taskfile" with the name of the task file. Put the name of the task file in quotation marks. For example,

```
    t:<all: call sa-t in "ULTRA.2">!
```

Ultra 4 returns control to the original taskfile and macro after running the macro. This is equivalent to calling a macro subroutine, except that you can store the macro outside the current macro table.

The new <call> command takes between 10-24 bytes (depending on the length of the taskfile name), so use the command to call larger macros (those with 30 or more tokens or text characters) to save bytes in the macro table.

Once you move one or more macros to another task file, recompile your original macro set. Repeat these procedures with more macros if the macro set is still too large for the compiler.

Thus, the new <call> command lets you divide your macros into two or more task files. That lets you create large and complex macros which Ultra 4 can run as though they were all in memory simultaneously.

Figure 6: Convert <getvar> for Ultra 4

UltraMacros 3.1

```
x:<all:
  getvar 2:              { use set 2 }
  $1 = "String #1":
  $2 = "String #2":
  X = 5:
  putvar 2:              { store set 2 again }
>!
```

Ultra 4

```
x:<all:
  $21 = "String #1":     // store in $21
  $22 = "String #2":
  X(2) = 5:              // store in X(2)
>!
```

var> commands; you will have to modify all macros that use these commands. This involves following the logic of the macro to determine what to use for each variable name.

Here is a suggestion to help with that conversion: Use the old value of <getvar> as a subscript for numeric variables and as the tens value of string variables. See the example in *Figure 6*.

Conclusion

This article describes how to convert your UltraMacros 3 macros to run under Ultra 4. The remaining articles in this series will describe how to use the new features of Ultra 4 to create exciting new macros and improve your existing macros.

*[Mark Munz, the author of **The UltraMacros Primer**, Companion Plus, TimeOut TextTools, MacroEase, Late Nite Patches, and other AppleWorks enhancements, is a programmer with Beagle Bros.]*

Need answers to your AppleWorks questions?

Call **NAUG**'s Members Helping Members volunteers listed in the ***AppleWorks Forum*** for the help you need.

News and Special Offers for NAUG Members

NAUG

As the nation's largest Apple II user group, **NAUG** continues to encourage member-to-member support services.

Judging from our mail, the **NAUG** membership includes a significant number of sexagenaria, septuagenaria, and octogenaria.

At the request of these members, **NAUG** recently launched a "Seniors Helping Seniors" program. All **NAUG** members interested in sharing the camaraderie of our creative, active, and interesting senior members qualify for this program. We welcome volunteers of any age; please send your name, address, daytime and evening telephone numbers, and whether you are a beginner, intermediate, or advanced AppleWorks user to "Seniors Helping Seniors", **NAUG**, Box 87453, Canton, Michigan 48187. We will publish the list of volunteers in a future issue of the ***AppleWorks Forum***.

Apple Expo East

NAUG members attending Apple Expo East should bring a copy of the ***AppleWorks Forum*** and ask for their $5 **NAUG** discount on the $20 cost of admission.

Apple Expo East features presentations by Roger Wagner, Phil Shapiro, Marty Knight, and other well-known members of the Apple II community. More than 60 companies have booths at the show and many will offer special Expo prices on their products. The show is October 2 - 4 at the Park Plaza Hotel in Boston.

[Event Specialists, 17 Lilac Road, Sharon, Massachusetts 02067; (800) 955-6630; In Mass.: (617) 784-4531.]

Hurricane Andrew...
Our heartfelt concern for our **NAUG** colleagues who suffered so greatly in Hurricane Andrew. Please let us know how we can help you recover from this disaster.

DRIVE SAFE. . . *DRIVE HARD!*

Inside and out, we've got a drive for you. From our reliable 143K 5.25" floppy drive to our super-fast 100MB Vulcan Gold Drive, Applied Engineering's disk drives are driven to be the best.

Æ 3.5" Drive

400K/800K/1.44MB Floppy Drive

- Format, read and write 400K and 800K floppy disks on IIe, IIc, IIcPlus, IIGS, Mac and PC Transporter
- ÆHD version gives you 1.44 megabytes on Macintosh and on IIGS with Apple's FDHD Controller Card
- Fully Apple compatible
- 2-way LED indicator, green for read, red for write
- Auto eject
- Daisy chain port—works both ways, connect a drive to it, or connect it to another drive
- Stack and footprint compatible with Apple Drive
- Conserver compatible
- Full One-Year Warranty

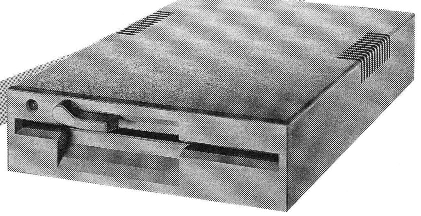
Æ 3.5" Drive (800K) $279
ÆHD 3.5" IIGS High Density Drive (1.44MB)....................................$339

Vulcan Gold

40 to 100MB Hard Drive

- Designed specifically for the Apple II series
- Preformatted—plug it in and you're off!
- Vulcan GS preformatted with ***System 6.0***
- Recognizes ***ALL*** Apple software including GS/OS, ProDOS, DOS 3.3, Apple Pascal, CP/M and MS-DOS for the PC Transporter
- Can be split in up to 16 different partitions
- Easily controlled through Control Panel Device
- 70 watt power supply (nearly twice the capacity of competing systems)
- Super-fast, state-of-the-art, IDE caching drive
- Tested and proven for years with all existing peripherals
- Support for years to come
- Full One-Year Warranty

40MB Vulcan Gold Drive...............$899
100MB Vulcan Gold Drive..........$1,795



5.25" Drive

143K Floppy Drive

- 143K Formatted, 35 Tracks, 16 Sectors
- Fully compatible with the IIe Emulation Card for the Macintosh LC
- Fully compatible with all Apple II models
- Proven Reliability
- Super Quiet
- Available Now!
- Full One-Year Warranty

5.25" Drive..................................$210

Order today! To order or for more information, see your dealer or call (214) 241-6060 today, 8:30 AM to 6:30 PM (CST), M–F. Or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add applicable sales tax.

ÆE

APPLIED ENGINEERING®
The Apple enhancement experts since 1979.
A Division of Æ Research Corporation

(214) 241-6060
FAX: (214) 484-1365

3210 Beltline
Dallas, TX 75234

Made
★ IN THE ★
USA

Play "AppleWorks Jigsaw Puzzle"

by Phil Shapiro

This is the first of two articles that describe how to use AppleWorks to create reading comprehension exercises. This month, Mr. Shapiro describes how to create "jigsaw puzzles". Next month, Mr. Shapiro describes his game of AppleWorks Anagrams.

Several years ago, I was smitten by an educational language arts game named *Hide 'n' Sequence*, published by Sunburst. The game starts by presenting the opening sentence from a reading passage. You then reconstruct the entire passage by successively choosing from three possible sentences. You can take as much time as you need to decide which of the three sentences logically follows from the preceding material. This type of language exercise helps students develop an appreciation for the subtle connections that link sentences into meaningful writing.

Of course, you can create your own variation of *Hide 'n' Sequence* with AppleWorks. This game has all the fun and excitement of a physical jigsaw puzzle except, instead of fitting together pieces of wood and cardboard, students fit together sentences and paragraphs into a unified, meaningful whole. And your students will learn how to use AppleWorks as they play.

Pairs or triads of students can play, but the most exciting approach has the entire class work together to create and solve one puzzle. You will need to customize my suggested procedures to allow for differences in class size and computer set-ups.

Students need a mature appreciation of the subtleties of language to enjoy and benefit from this game. You will find it difficult to use the game with children below sixth grade.

How It Works

Here is how the game works. First you find or create one or more fiction or non-fiction passages whose sentences progress logically from the beginning to the end of the passage. I suggest that you try this activity first with non-fiction passages; those passages often have an inherently tight logical structure.

Mixing up the AppleWorks jigsaw puzzle is almost as much fun as trying to solve it. To add drama to this exercise, you might designate two or three students as a "detective team". Send the "detectives" out of the room while the class reorganizes the passage; the secrecy adds suspense to the exercise.

First you load a reading passage into AppleWorks. Then have one or more students in your class use AppleWorks' Move Command to move one sentence or an entire paragraph from the middle to the end of the passage. However, do not move text from the opening sentences of the passage; the "detectives" need a "setting" to help them understand the moved sentence or paragraph. Try to move text that contains subtle but logical links to the body of the prose.

> "This exercise helps students appreciate the subtle connections in meaningful writing."

Then call the detectives back into the classroom, print the passage, and give them a printed copy to study. Tell the detectives to read the moved sentence or paragraph before trying to figure out where to put it.

Let the detectives confer among themselves looking for subtle clues. When they think they solved the puzzle, they can go to AppleWorks and use the Apple-M command to move the sentence or paragraph back to its proper position.

You can add to the fun by turning off the large screen monitor while the students reorganize the paragraph. Then let the detectives use TimeOut FileMaster or the freeware File Compare utility to compare the reconstructed AppleWorks file with the original. The students can run the file comparison and turn on the monitor with a flourish to show the results.

Help Getting Started

To help you get started, I assembled several passages that **NAUG** will distribute on its new "Shapiro's Jigsaw Puzzle Disk". The disk also includes a bootable copy of File Compare and some instructions to help you use the program. Several of these passages also appear on the new Big Text Machine Sample Files disk. *[Ed: Mr. Shapiro's Big Text Machine lets you display AppleWorks and other text files in attractive large fonts on your Apple II screen.]*

Not all teachers will have the time or inclination to create their own reading passages for "AppleWorks Jigsaw Puzzle". But if several of the more energetic **NAUG** teachers each contribute two or three original reading passages, we should be able to assemble several more **NAUG** disks for use with the game.

It may even be possible to coax high school students to write reading passages for use by younger students in this activity. Not only would such writings be useful to other teachers and students, but the authors of these passages would have a ready "writing sample" for use at job interviews. Not to mention extra "brownie points" on their resumes.

[Phil Shapiro is the founder of Balloons Software, 5201 Chevy Case Parkway, NW, Washington, DC 20015. Balloons Software developed Big Text Machine, Number Squares, and other Apple II educational software. You can reach Mr. Shapiro on GEnie as "p.shapiro1" or on America Online as "pshapiro".]

["Shapiro's Jigsaw Puzzle Disk" costs $4 (5.25-inch disk) or $6 (3.5-inch disk) plus $2 s/h per order from the NAUG Public Domain Library.]

News and Special Offers for NAUG Members

Bemak Enterprises

Bemak Enterprises offers **NAUG** members special prices on the company's AppleWorks 3.0 Golf, Handicap Golf, and Bowling Scores templates.

Bemak's *Golf Scores Disk* contains word processor and spreadsheet templates designed to help non-handicap golfers keep track of their scores and statistics. Other templates print a score card you can use on the course and different award certificates.

Golf Scores tracks up to seven 9-hole rounds of golf on 128K systems and twenty-eight 9-hole rounds on 256K computers.

Bemak's *Handicap Golf Disk* makes it easy to calculate your USGA handicap by tracking your performance on your fifty most recent 18-hole rounds of golf. The Handicap Golf template can compute your course-specific handicap if you provide the USGA Course Rating and the USGA Slope Rating for the course.

Other templates on the disk track many of the same data and produce the same score card and certificates as the *Golf Scores Disk.*

The company's *Bowling Scores Disk* tracks individual and team bowling scores and statistics. Templates track the number of strikes and spares rolled per game and compute the averages and handicaps for individual and team bowling. The disk includes an AppleWorks data base file that can help a league secretary record information on all bowlers.

Word processor templates on the disk print score sheets you can use at the bowling alley and award certificates you can distribute to the best performers.

Each Bemak disk lists for $19.95. Until January 1, 1993 **NAUG** members can buy each disk for $12.95 plus $2.50 s/h per order. Specify 3.5-inch or 5.25-inch disks and include your **NAUG** membership number with your order.

[Bemak Enterprises, 728 Royal Street, Alton, Illinois 62002; (618) 462-7419.]

A Gradebook Template - Part One

by Stan Hecker

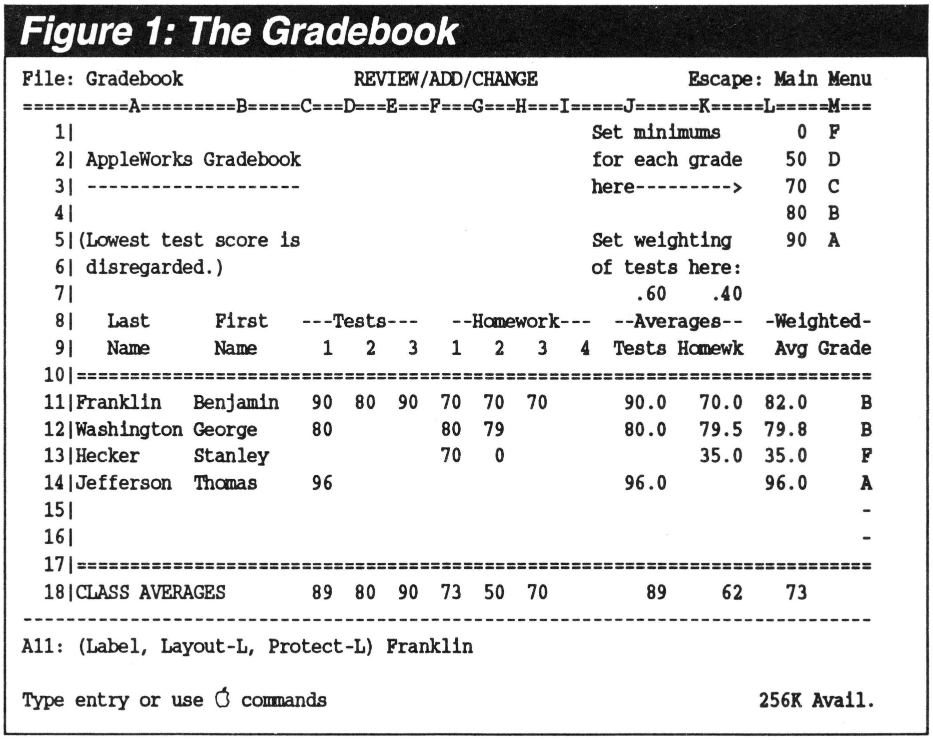
This is the first of two articles that describe how teachers can use AppleWorks to maintain and report student grades. The author assumes that you know the basics of spreadsheet operation. NAUG members not familiar with the AppleWorks spreadsheet should first read the articles in the booklet entitled "How to Get Started with the Spreadsheet Module", which costs $9.25 postpaid from NAUG.

Show me five teachers and I will show you five different recordkeeping systems. Although there are probably as many different gradebooks as there are teachers, this month's gradebook template (see *Figure 1*) includes some ideas that you can use in your electronic recordkeeping system.

A working copy of the template appears on this month's issue of **NAUG on Disk**.

This is the first of two articles about classroom gradebooks. This month you will learn how to create the template. Next month you will use this template to produce individual reports that use the close integration of the three AppleWorks modules.

Figure 1: The Gradebook



```
File: Gradebook                REVIEW/ADD/CHANGE              Escape: Main Menu
==========A=========B=====C===D===E===F===G===H===I=====J======K=====L=====M===
   1|                                               Set minimums       0  F
   2| AppleWorks Gradebook                          for each grade    50  D
   3| --------------------                          here--------->    70  C
   4|                                                                 80  B
   5|(Lowest test score is                          Set weighting     90  A
   6| disregarded.)                                 of tests here:
   7|                                                   .60    .40
   8|   Last      First   ---Tests---   --Homework---  --Averages--  -Weighted-
   9|   Name      Name      1   2   3   1   2   3   4  Tests Homewk   Avg Grade
  10|==========================================================================
  11|Franklin   Benjamin   90  80  90  70  70  70       90.0   70.0  82.0     B
  12|Washington George     80          80  79           80.0   79.5  79.8     B
  13|Hecker     Stanley                70   0                  35.0  35.0     F
  14|Jefferson  Thomas     96                           96.0         96.0     A
  15|                                                                         -
  16|                                                                         -
  17|==========================================================================
  18|CLASS AVERAGES        89  80  90  73  50  70         89     62    73
-------------------------------------------------------------------------------
A11: (Label, Layout-L, Protect-L) Franklin

Type entry or use  commands                                        256K Avail.
```

This template can accommodate up to six students and five measurements, which is too small for practical use. The section entitled "Expanding this Template" near the end of the article describes how to enlarge the template to accommodate more students and grades.

The template refers specifically to "tests" and "homework", although you can use the template to store grades from any project or assignment.

Much of the complexity of the formulas in this gradebook comes from the decision to drop the lowest test score from each students' average. Beginning spreadsheet users can enter the formula without understanding its logic. Advanced spreadsheet developers might enjoy trying to create the formula in cell J11 before they get to step #14 in this article. Then compare that formula to the simpler formula in cell K11.

The formulas in the template also demonstrate how you can display blank cells instead of error messages. That trick uses AppleWorks 3.0's ability to manipulate text in the spreadsheet. Earlier versions of AppleWorks did not offer this capability, so the template requires AppleWorks 3.0.

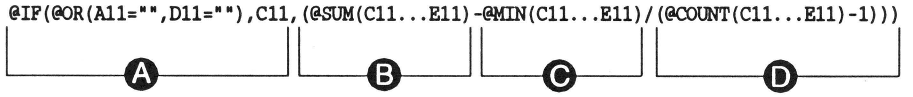
Building the Template

1. Start by launching AppleWorks and creating a new spreadsheet called "Gradebook". Save the template to disk often as you work.

Figure 2: Format for Cells

Cells	Type	Layout
The block A8 through B9	Label	Centered
The block A11 through B16	Label	Left Justified
J7 and K7	Value	Fixed, 2 decimal places
L1 through L5	Value	Fixed, Ø decimal places
M1 through M5	Label	Centered
The block J11 through L11	Value	Fixed, 1 decimal place

Figure 3: Formula for Cell J11



```
@IF(@OR(A11="",D11=""),C11,(@SUM(C11...E11)-@MIN(C11...E11)/(@COUNT(C11...E11)-1)))
```

2. Widen column A to 12 characters to accommodate long last names.
3. Narrow column B to 8 characters.
4. Narrow columns C through I to 4 characters.
5. Narrow columns J and K to 7 characters.
6. Narrow columns L and M to 6 characters.
7. Issue an Apple-V command and set the standard value format to “Fixed, Zero Decimal Places” and set the standard label format to “Right Justified”. Also set the calculation frequency to “Manual”.
8. Use the Apple-L command to set the format for the cells listed in *Figure 2*.
9. Add the labels in rows 2 through 10 and 17 through 18 in *Figure 1*. (Remember to enter a quotation mark to start a label that begins with a hyphen, an equals sign, or a blank space.)
10. Enter the formula 1-J7 in cell K7. The number “1” will appear in the cell. This formula automatically calculates the weight assigned to student homework after you enter the weight assigned to test scores.
11. Enter the value of zero in cell L1.
12. Enter the value 49.5 in cell L2. A “50” will appear in your template. Similarly, enter 69.5 in cell L3, 79.5 in L4, and 89.5 in L5. These values insure that students who get grades up to 1/2-point below the cut-offs receive the next higher grade.
13. Enter the letter grades F, D, C, B, and A in cells M1 through M5 respectively.

The Formulas

Now you will enter the formulas in row 11. The pairs of quotation marks ("") in these formulas signify a “nothingness”, technically called a “null set”. (To a computer, a blank space is a character and a zero is a value. Thus, blanks and zeros are “something”. Putting "" in an equation signifies that you want to test for “emptiness”. For example, the formula `@IF(A1=Ø,"",B1)` leaves the current cell empty if cell A1 contains a zero or non-numeric entry.)

First you will enter the formula in *Figure 3* into cell J11. (Do not enter the formula until you read the step-by-step instructions below. The formula is too long to fit on the data entry line.)

This formula says:

Segment Ⓐ : “If the last name cell is blank, then there are no grades in this row; leave cell J11 blank.

“Or, if there is no second test score in cell D11, display the student’s first test score as the average score.”

Segment Ⓑ : If both A11 and D11 contain entries, this row contains data for a student who took at least two tests. Now you need to throw out the lowest test score. Segment B of the formula says: “Sum the test scores in cells C11 through E11.”

Segment Ⓒ : “Then subtract the single lowest score in that range from the sum.”

Segment Ⓓ : “Now divide the result by the total number of test scores in the same range, less one, to account for the single test score you threw out. Display this average.”

As indicated earlier, the formula will not fit on the data entry line. The following steps demonstrate a work-around that lets you enter long formulas. The technique is to enter a portion of the formula, then display the formula on the Edit Line and enter the remainder of the formula. Follow these steps:

14. Enter the following into cell J11. Then press the Return Key.

```
@IF(@OR(A11="",D11=""),C11,(@SUM(C11.E11)
-@MIN(C11.E11)/1))
```

15. Issue an Apple-U command. AppleWorks will display the following formula on the Edit Line at the bottom of the screen:

```
@IF(@OR(A11="",D11=""),C11,(@SUM(C11...E11)
-@MIN(C11...E11)/1))
```

Note that AppleWorks substituted three dots for the single periods you typed between "C11" and "E11" at two places in the formula. Start by using the Arrow Keys and the Delete Key to delete the extra periods so you can fit the final formula on the Edit Line.

16. Now put the cursor at the end of the formula and press the Delete Key three times to delete the two right parentheses and the digit "1" from the end of the expression. The formula will look like this:

```
@IF(@OR(A11="",D11=""),C11,(@SUM(C11.E11)
-@MIN(C11.E11)/
```

with the cursor blinking at the end of the formula.

17. Finish the expression by typing the following after the "/":

```
(@COUNT(C11.E11)-1)))
```

Then press the Return Key.

That completes the formula in *Figure 3*, which, as indicated earlier, is too large to fit on the screen.

18. Enter the following formula in cell K11:

```
@IF(@OR(A11="",F11=""),"",@AVG(F11.I11))
```

This formula says "If the last name cell is blank or if the first homework field is blank, then there is either no student here or the student did not hand in any homework. In either case, display a blank. Otherwise, display the average of the homework assignments."

Unlike cell J11, which required a lengthy formula, your decision to retain all of the student's homework scores lets you use the simpler @AVG function in cell K11.

19. Use the Apple-C command to copy the formulas in cells J11 and K11 into rows 12 through 16. Issue an Apple-R to command "relative" for all cell references.

20. Enter the following formula in cell L11:

```
@IF(A11="","",@IF(@OR(J11="",K11=""),
J11+K11,(J11*J7)+(K11*K7)))
```

This formula computes the weighted average. The formula says, "If the last name cell is empty, then leave this cell blank. Otherwise, check both the 'test average' and 'homework average' cells. If either the 'test average' or 'homework average' cell is blank, display the sum of both fields here. If both cells contain values, multiply each by the weighting factors near the top of this column, add them together, and display the result."

21. Enter this formula in cell M11:

```
@IF(A11="","-",@LOOKUP(L11,L1...L5))
```

This formula assigns a letter grade for each student. It says: "If the last name cell is blank, display a dash as a placeholder. Otherwise, look up the student's grade in the table in cells L1 through M5."

22. Copy cells L11 and M11 into rows 12 through 16. Respond to the "Relative or No change?" question with "Relative" *except* when AppleWorks highlights cells J7, K7, L1, and L5. In those four cases, press "N" to specify "No change".

23. Now you will enter the formula that calculates the class average for each test and assignment.

Type this formula into cell C18:

```
@IF(@COUNT(C1Ø...C17)=Ø,"",@AVG(C1Ø...C17))
```

This formula counts the number of entries in cells C10 through C17 and displays a blank instead of an error message if it finds no student grades in those cells. If it finds one or more student grades, the formula displays the average of those grades.

24. Copy the formula from cell C18 into cells D18 through L18. Issue an Apple-R to specify "Relative" in response to the "Relative or No change?" question.

Protect Your Work

25. Put the cursor in cell A1, issue an Apple-L, select "Block", and set protection for the whole spreadsheet to allow "Nothing".
26. Use the Apple-L command to set the protection for cell J7 and cells L1 through L5 to "Values only".
27. Set protection for the block of cells A11 through B16 to "Labels only".
28. Set protection for the block of cells C11 through I16 to "Values only".
29. Use the Apple-V command to turn protection on and to change the calculation frequency to "Automatic".
30. For extra security, lock the template, using BASIC, Copy II+, FileMaster, or some other utility program. (For step-by-step directions, see the article entitled "How to Lock Your Templates" in the May 1991 issue of the ***AppleWorks Forum***.)
31. Save the template to disk; you will use it next month to explore the integration of the AppleWorks modules.

Using the Template

Follow these steps to use the template:

1. Load the template onto the AppleWorks desktop.
2. Issue an Apple-N command and name the file "Class 1".
3. Set the weighting so that your tests determine 50% of the students' final grade by entering ".5" into cell J7.
4. Change the standard for passing to 65% by entering "64.5" into cell L2.
5. Enter student names and their scores on your tests and assignments.

Remember that you can use the Apple-A command to arrange the student records alphabetically by name, by overall class standing, or by any test or assignment. Just remember to save your work before re-arranging the data; AppleWorks does not offer an "unarrange" command.

Expanding the Template

Unfortunately, few of us get to teach classes with only six students. Follow these steps to add students to the template:

1. Issue an Apple-V command and set Protection to "No".
2. Put the cursor anywhere in row 17 and issue an Apple-I command. Indicate that you want to insert rows and specify the number of rows to insert.
3. Put the cursor in cell A16 and use the Apple-C command to copy cells A16 through K16 into the new rows. Issue an Apple-R in response to the "Relative or No change?" question.
4. Use Apple-C to copy the formula from cell L16 into the new rows. Respond "Relative" when AppleWorks highlights "A16", "J16", and "K16". Respond "No change" when AppleWorks highlights "J7" and "K7".
5. Use Apple-C to copy the formula from cell M16 into the new rows. Respond "Relative" when AppleWorks highlights "A16" and "L16". Respond "No change" when AppleWorks highlights "L1" and "L5".
6. Use Apple-V to re-establish cell protection.

Adding Tests and Homework Assignments

Adding tests and homework assignments involves inserting columns into the template. Add the columns in front of the last test and assignment. That eliminates the need to edit the formulas in columns K and L of the basic template.

Follow these steps to add more assignments:

1. Put the cursor in cell I1, issue an Apple-I, and insert as many extra columns as you need in your gradebook.

2. Move the cursor back to cell H1. Use Apple-C and Apple-9 to copy the entire column H into the new columns. Press Apple-R to command "Relative" in response to the "Relative or No change?" question.
3. Correct the copied labels in row 8 for the cells. Also correct the assignment numbers at the top of the columns.

Follow these steps to add space for more tests:

1. Put the cursor in cell E1, issue an Apple-I, and insert as many extra columns as you need in your gradebook.
2. Move the cursor back to cell D1. Use Apple-C and Apple-9 to copy the entire column D into the new columns. Press Apple-R to command "Relative" in response to the "Relative or No change?" question.
3. Move to row 8 and clean up the copied labels for the cells. Similarly, change the test numbers at the top of the columns.

Save the revised template to disk under its new name. Do not discard the original version; you will use this template next month to explore the integration of the AppleWorks modules.

[Stan Hecker is on the administrative staff at Michigan State University, East Lansing, Michigan, and is a partner in H&H Consulting, a Michigan concern specializing in school district financial and population analyses.]

[The author thanks to Dr. Avram Malkin, an Assistant Professor at the DeVry Technical Institute (Woodbridge, NJ) for submitting a gradebook template that inspired these articles. His incisive suggestions about integrating the modules served as the foundation for next month's discussion.]

[A working copy of this template appears on this month's issue of ***NAUG on Disk****, which costs $10 from* ***NAUG****. The template requires AppleWorks 3.0;* ***NAUG on Disk*** *requires a 3.5-inch disk drive.]*

News and Special Offers for NAUG Members

JEM Software

JEM Software announced special **NAUG** prices on its line of AppleWorks 3.0 data base enhancements. Until December 1, **NAUG** members who order TotalControl 2.1 for the regular $60 price will receive a copy of DoubleData 2.0 (regularly $40) free. **NAUG** members who order TotalControl can also buy the company's DB Pix enhancement for only $12 (regularly $25).

TotalControl 2.1 adds many features to the AppleWorks data base module, including relational capabilities and data entry rules. (A comprehensive review of TotalControl appears in the May 1992 issue of the ***AppleWorks Forum***.)

DoubleData 2.0 seamlessly expands the AppleWorks 3.0 data base to include up to 60 categories per record.

DB Pix 1.0 lets you display Single HiRes, Double HiRes, and PrintShop graphics in your data base layouts.

TotalControl, DoubleData, and DB Pix come on 3.5-inch disks unless you request 5.25-inch copies of the programs. Include your **NAUG** membership number with your order. Add $3 s/h ($5 overseas); Colorado residents must add sales tax.

[JEM Software, 7578 Lamar Court, Arvada, Colorado 80003; (303) 422-4856 for 24-hour Visa/MasterCard order line or fax.]

Classifieds

How to Use UltraMacros Subroutines

by Keith Johnson

Understanding how a macro operates is a good idea, even if you do not create your own macros. You can use these skills to improve the macros you use in your daily operations.

Macro programmers often use subroutines; separate macros used by other macros to accomplish a particular task. You can identify the subroutine macro by the <asr> ("AppleWorks Subroutine") designator that comes right after the keystroke that names the macro. A subroutine often saves valuable space in your macro set.

Waiting for a Keypress

For example, imagine that you want a macro to pause while displaying a message and wait for the user to press a key. You may want to store the value of that key for further use, then clear any message on the screen. You can accomplish that task by including the following lines in your macros:

```
k = key :     { Get a keypress.                    }
msg "" :      { Clear the current message, if any. }
$1 = chr$ k : { Define a string variable that      }
              { contains the keypress.             }
```

This does the job. But suppose that several macros need to perform this task. If you insert these commands into each macro, you will use up valuable space. Instead, you should define the following subroutine macro:

```
K:<asr><   { Define subroutine.                    }
 k = key : msg "" : $1 = chr$ k>!
```

Then call this subroutine whenever you need to capture a keystroke. For example:

```
D:<all>
 <msg ' today is ' + date + ' (key) ' : { Display date. }
 sa-K>!         { Call sa-K.                            }
```

This macro displays the date on the message line and then runs the <sa-K> subroutine, which waits for a keypress and clears the message. Note that this macro does not use the value stored in $1 by the subroutine.

Layers

UltraMacros' ability to call a subroutine from within a subroutine gives you even more flexibility. For example, imagine that you want some of your macros to clear the message without waiting for a keypress. You want other macros to wait for the keypress. The best approach is to separate the sa-K subroutine into two subroutines, one of which calls the other, as follows:

```
M:<asr><msg "">!       { Clear the message.         }

K:<asr><k = key :      { Get a keypress.            }
 sa-M :                { Call sa-M to clear message.}
 $1 = chr$ k>!         { Get string variable.       }
```

You could even create a third subroutine that does not define $1:

```
N:<asr><k = key :      { Get a keypress.            }
 sa-M>!                { Call sa-M to clear message.}
```

Of course these examples divide the tasks into overly-small chunks, but I want to keep things simple for explanatory purposes.

Keep Things in Order

Make certain that the calling macro calls the correct subroutine. This can be a problem if you use the same defining token, or name, for your subroutine that you used for a "regular" macro. For example, consider this set of three macros:

```
K:<asr><k = key :              { Get a keypress.    }
    msg "">!                   { Clear the message. }
```

```
K:<awp><$5 = "Date is " + date :
 msg $5>!                { Display the date message. }

M:<awp><msg "Press a key" :     { Display a message. }
 sa-K>!                         { Call sa-K.         }
```

If you press <sa-M> in the word processor, the <sa-M> macro displays the message "Press a key" and then calls <sa-K>. UltraMacros goes to the top of your macro set and starts looking for a macro named <sa-K> that it can run from the word processor. In this case, it encounters the <sa-K> keypress/message-clear subroutine and it runs that macro, which is what you want.

But suppose you typed the same three macros in the following order:

```
K:<awp><$5 = "Date is " + date :
 msg $5>!                { Display the date message. }

K:<asr><k = key : msg "">! { Get a keypress.         }
                           { Clear the message.      }

M:<awp><msg "Press a key" : { Display a message.     }
 sa-K>!                     { Call sa-K.             }
```

Pressing <sa-M> still displays the "Press a key" message and sends UltraMacros looking for a <sa-K> macro. But now it encounters the date macro first, so it replaces the previous message with a new one proclaiming the current date. That is not what you intended.

The best way to insure that subroutines run properly is to put them at the beginning of your macro set. Since you cannot run subroutines by pressing keys, you cannot run these macros accidentally. You can even assign the subroutines the same name that you use for the macro that calls them.

Using Ultra 4

Ultra 4 gives you two ways to handle subroutines. First, the new version of UltraMacros continues to support the <asr> macros described above. Ultra 4's expanded <call> command also lets you store your subroutines in a separate task file that you can call from any macro. *[Ed: See the UltraMacros Primer article that starts on page 11 for examples of how to use Ultra 4's new <call> command.]*

Second, Ultra 4 supports "labels", which consist of a sequence of actions that you can call from any macro. For instance, instead of

```
K:<asr><k = key : msg "" : $1 = chr$ k>!
```

you can define the label

```
#clearmessage =   k = key : msg "" : $1 = chr$ k
```

and insert "#clearmessage" anywhere in a macro set where you previously used <sa-K>.

However, you must respect the following two limitations in your labels:

1. All label definitions must fit on a single line, so you cannot use labels to define complicated procedures.
2. You must define all your labels at the beginning of your macro set. That is a good practice anyway.

[Keith Johnson is Associate Director of the Fleischmann Planetarium at the University of Nevada.]

How to Enhance Your Documents with MouseText - Part Two

by Warren Williams and F. David Stephansky

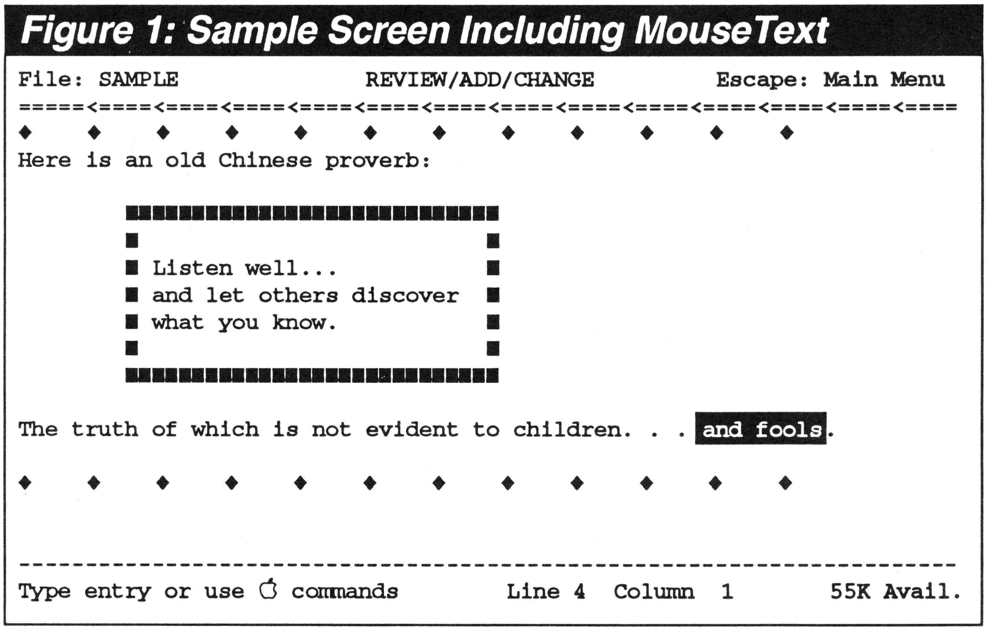
This is the second of two articles that describe how to add MouseText to your AppleWorks word processor documents. The authors assume that you read the article in last month's issue of the ***AppleWorks Forum.***

Last month you learned how to incorporate MouseText in your printed output. The procedures we described let you produce attractive documents that include boxes, diamonds, and other graphic elements.

Of course, much of today's writing never appears on a printed page. Increasingly, we design documents that users read on the computer screen. For example, program developers often provide directions in word processor files on a disk. Teachers give students electronic copies of their instructions and comments. And other documents, such as **NAUG on Disk** and TimeOut Central, only exist on disk and never appear on a printed page.

As you saw last month, the techniques you use to add MouseText to printed documents actually *decrease* their on-screen readability. This month you will learn how to prepare attractive documents that display MouseText, underlining, and inverse characters on the screen. You can use these techniques to develop attractive screen displays like the example in *Figure 1.*

Figure 1: Sample Screen Including MouseText



```
File: SAMPLE                REVIEW/ADD/CHANGE           Escape: Main Menu
=====<====<====<====<====<====<====<====<====<====<====<====<====<====
◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆
Here is an old Chinese proverb:

        ■■■■■■■■■■■■■■■■■■■■■■■■■■■■
        ■                          ■
        ■ Listen well...           ■
        ■ and let others discover  ■
        ■ what you know.           ■
        ■                          ■
        ■■■■■■■■■■■■■■■■■■■■■■■■■■■■

The truth of which is not evident to children. . . and fools.

◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆    ◆


----------------------------------------------------------------------
Type entry or use  commands         Line 4  Column  1        55K Avail.
```

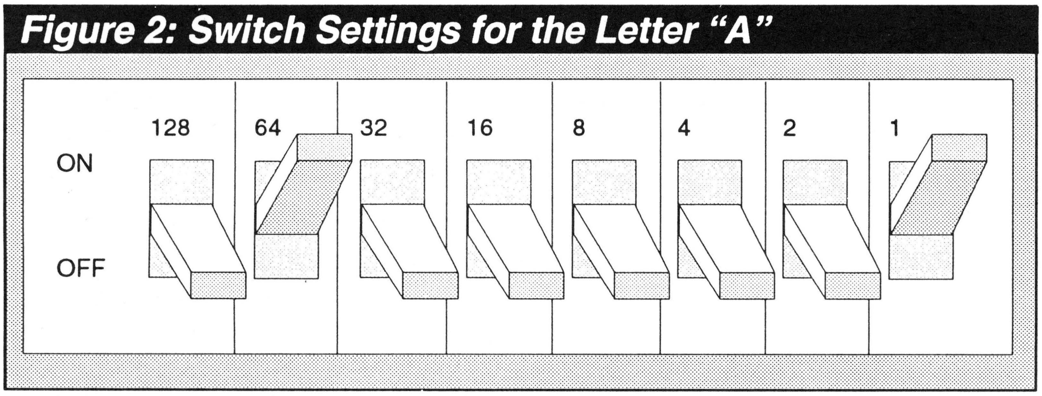
Some Background

Here is some background to help you understand the procedures that generate the on-screen MouseText.

Computers store each character in one "byte" of memory. Each byte includes eight storage locations (called "bits") that you can think of as switches. The computer represents each letter of the alphabet by a different combination of on-off settings of these "switches". For example, *Figure 2* presents a schematic representation of the "switch settings" stored in your computer when you type the letter "A".

You can arrange these switches in 256 different combinations of on-off patterns. Thus, each set of eight switches can represent 256 different characters. However, you can only type about 128 different printable characters from the keyboard. As a result, your computer never needs the left-most switch (which represents "128", the highest number in the switch sequence), to store a text charac-

Figure 2: Switch Settings for the Letter "A"



ter. Thus, the industry standard for text characters (called the ASCII standard) only uses the lowest seven of the eight switches to represent different characters.

Of course, computer manufacturers like to use the full capabilities of their systems, so they use this eighth (or "high") bit to perform other functions. Apple IIc, IIc Plus, IIGS, and enhanced Apple IIe computers use this bit to tell the computer whether to use the standard character set (high bit turned off) or an alternate character set built into these computers. The alternate character set includes inverse versions of all the lower-case characters and a set of graphic "characters" that we call MouseText.

Obviously, then, the way to produce MouseText and inverse characters on the screen is to type the keyboard equivalent of each character and then turn the high bit switch on for each of those characters.

Turning the High Bit On

Unfortunately, there is no "turn-the-high-bit-switch-on" key on your keyboard; you must use software to change the switch settings.

Advanced Apple II users can use "disk zapping" programs such as those included with ProSel and Copy II+ to turn the high bit on for these characters. But that is a tedious process that must be done with care so you avoid changing any other characters in the file.

An easier approach is to type your document and then use a macro or TimeOut application to change these characters. We use Barclay Clemesha's "MouseText Maker", a TimeOut application included on the "Clemesha's AppleWorks Accessories Disk" available from **NAUG.** MouseText Maker makes it easy to display MouseText graphics, inverse characters, and underlined characters in AppleWorks documents on your screen.

Using MouseText Maker

Using MouseText Maker is a three-step process:

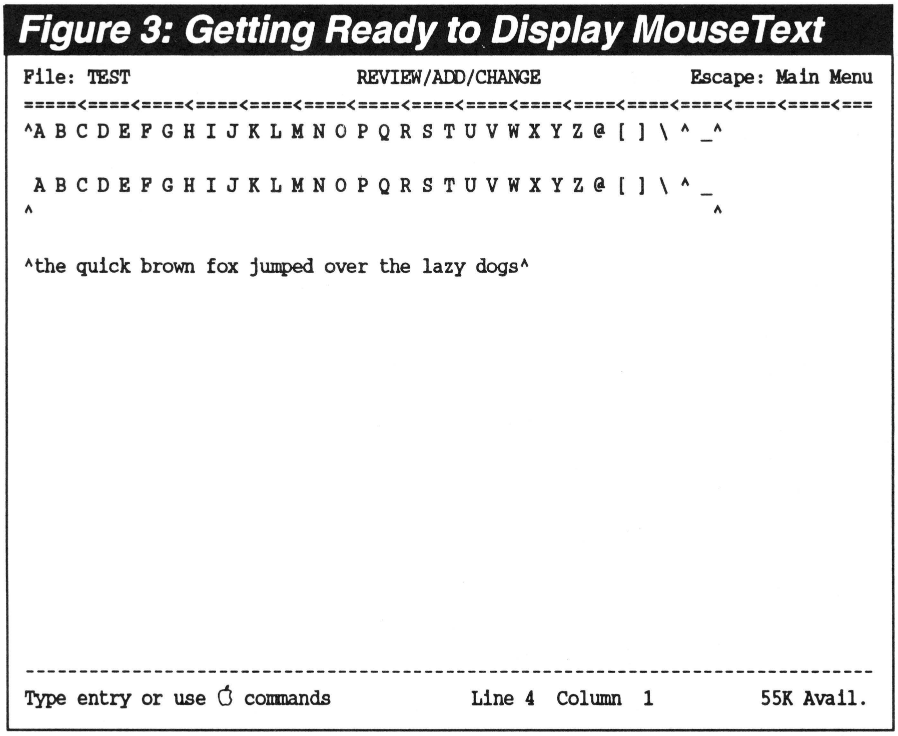
1. Install the program.
2. Create AppleWorks documents with embedded codes to turn MouseText, inverse, and underline on and off.
3. Run MouseText Maker to convert the characters.

The process is easy because (a) you only install MouseText Maker once, and (b) MouseText Maker uses Boldface Begin/End Commands to turn MouseText on and off and Underline Begin/End Commands to control underlining.

Just follow these steps:

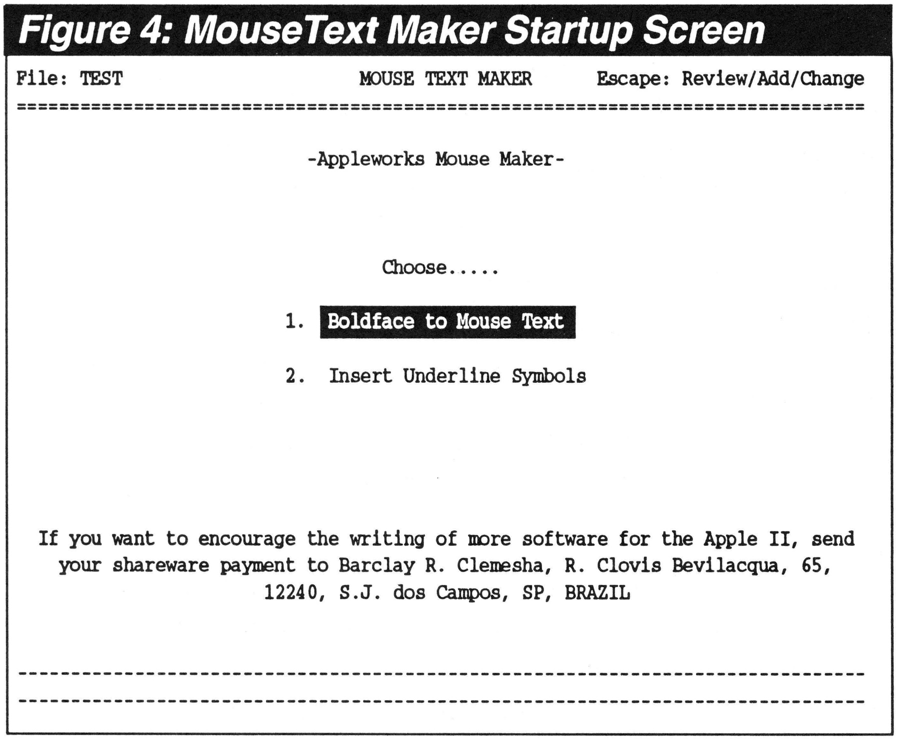
1. Install MouseText Maker by copying the file TO.MOUSE from the Clemesha's AppleWorks Accessories Disk onto your TimeOut Applications Disk or into the subdirectory you use to store your TimeOut applications.
2. Launch AppleWorks.
3. Prepare a new word processor document named "TEST" and set the left and right margins to Ø.Ø inches.
4. Type all the letters of the alphabet in upper case with a space between each letter.
5. Type the following characters at the end of the line with a space between each character: "@", "[", "]", "\", "^", "_".
6. Copy the entire line to the clipboard. Then leave a blank line and copy the text from the clipboard.

Figure 3: Getting Ready to Display MouseText



```
File: TEST                    REVIEW/ADD/CHANGE              Escape: Main Menu
=====<====<====<====<====<====<====<====<====<====<====<====<====<====<====<===
^A B C D E F G H I J K L M N O P Q R S T U V W X Y Z @ [ ] \ ^ _^

 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z @ [ ] \ ^ _
^                                                                ^

^the quick brown fox jumped over the lazy dogs^










------------------------------------------------------------------------------
Type entry or use  commands              Line 4  Column  1           55K Avail.
```

Figure 4: MouseText Maker Startup Screen



```
File: TEST                     MOUSE TEXT MAKER        Escape: Review/Add/Change
==============================================================================

                          -Appleworks Mouse Maker-



                                 Choose.....

                          1.  Boldface to Mouse Text

                          2.  Insert Underline Symbols





   If you want to encourage the writing of more software for the Apple II, send
         your shareware payment to Barclay R. Clemesha, R. Clovis Bevilacqua, 65,
                        12240, S.J. dos Campos, SP, BRAZIL


------------------------------------------------------------------------------
------------------------------------------------------------------------------
```

7. Issue an Apple-1 to put the cursor at the beginning of the first line of text and issue a Boldface Begin Command. Move the cursor to the end of the line and issue a Boldface End Command.

8. Insert a blank space at the beginning of the second line of text. (MouseText Maker removes the caret (^) that marks the Boldface Begin Command in line 1 but does not move the text on that line. Putting the space at the beginning of the second line of text will align each letter of the alphabet with its corresponding MouseText character.)

Now you will enter the Underline Begin/End Commands. These commands go on the first blank line under the second row of characters.

9. Move the cursor to the beginning of the blank line under the second row of characters and issue an Underline Begin Command.

10. Press and hold down the Space Bar to move the cursor so it is under the first blank space after the underline character. Then insert an Underline End Command.

11. Press the Return Key twice, enter a Boldface Begin Command, and type "the quick brown fox jumped over the lazy dogs" in lower case characters. Then enter a Boldface End Command. Your screen should look like the example in *Figure 3.*

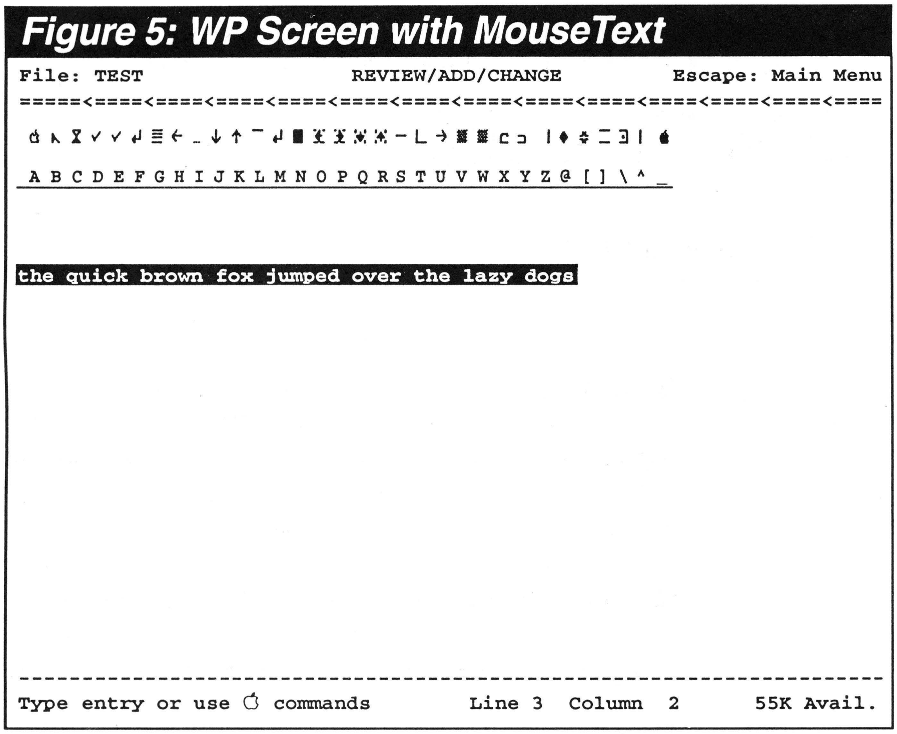
12. Go to the TimeOut Menu and launch MouseText Maker to display the screen that appears in *Figure 4.*

13. Press the Return Key to convert all the characters between the Boldface Begin and Boldface End Commands into MouseText.

14. Return to the TimeOut Menu, invoke MouseText Maker a second time, select #2 ("Insert Underline Symbols"), and press the Return Key. Your screen should look like the example in *Figure 5.*

To learn more about on-screen MouseText and underlining, we suggest that you repeat the tutorial in last month's article but issue Boldface Begin/End Commands in place of the Special Codes Commands described in that article. Then

Figure 5: WP Screen with MouseText



```
File: TEST                     REVIEW/ADD/CHANGE              Escape: Main Menu
=====<====<====<====<====<====<====<====<====<====<====<====<====<====<====<====

  ↖ ⧗ ✓ ✓ ↵ ≡ ← _ ↓ ↑ ‾ ↵ █ ⧗ ⧗ ⧗ ⧗ – └ → ▒ ▒ ⊏ ⊐  | ◆ ≑ ═ ⊐ | 

 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z @ [ ] \ ^ _


the quick brown fox jumped over the lazy dogs

--------------------------------------------------------------------------------
Type entry or use  commands           Line 3  Column  2          55K Avail.
```

launch MouseText Maker and convert your screen to the display that appears in *Figure 1*.

Tips and Ideas

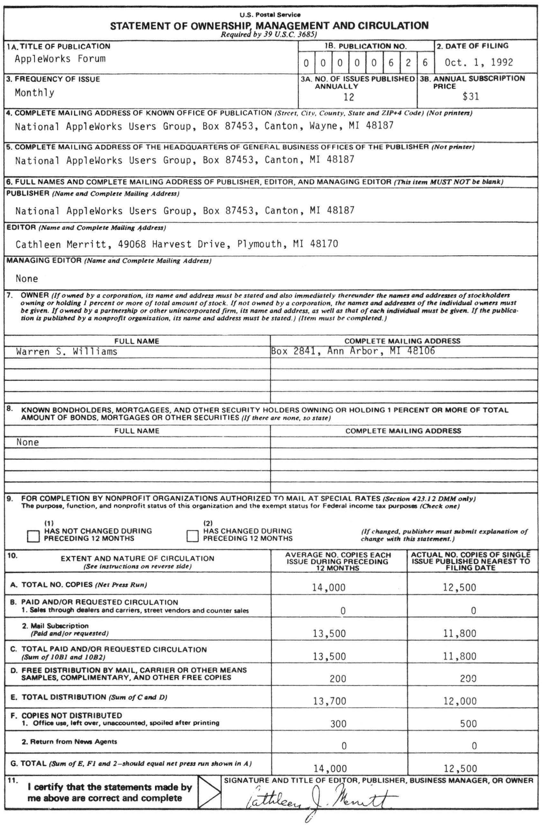
Here are some suggestions to help you use MouseText Maker:

1. MouseText Maker converts each upper case character to its MouseText equivalent. *Figure 3* in last month's article shows the MouseText equivalent for every character.
2. Use on-screen underlining sparingly, and only in documents that you double space on the screen. Since you must put the Underline Begin/End Commands on a blank line underneath the text you want underlined, you cannot use MouseText Maker to underline single spaced documents.
3. MouseText Maker is partially reversible. Insert Boldface Begin/End Commands around your MouseText and process the document through MouseText Maker a second time. The program will convert each MouseText graphic back to its corresponding upper case letter. You can use the Apple-D command to delete the underline from the screen.
4. Do not print your MouseText-ed documents. Printing MouseText involves using the procedures described last month to send MouseText Begin/End Commands to your printer. Although the on-screen and printed documents look similar, you must use different procedures to display and print MouseText.

[Dr. Warren Williams is on the faculty at Eastern Michigan University where he teaches courses in the Educational Technology program. He is the President of ***NAUG*** *and is a frequent contributor to the* ***AppleWorks Forum****.]*

[F. David Stephansky teaches computer courses for Fitchburg (MA) State College and the Plymouth County Education Association. You can reach Mr. Stephansky at 53 Simmons Ave., Whitman, Massachusetts 02382; (617) 447-5156.]

[Clemesha's AppleWorks Accessories Disk costs $4 (5.25-inch disk) or $6 (3.5-inch disk) plus $2 s/h per order from ***NAUG****. MouseText Maker is shareware; you send Dr. Clemesha $10 if you use the program after getting the disk from* ***NAUG****.]*



U.S. Postal Service

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION

Required by 39 U.S.C. 3685)

1A. TITLE OF PUBLICATION: AppleWorks Forum

1B. PUBLICATION NO.: 0 0 0 0 0 6 2 6

2. DATE OF FILING: Oct. 1, 1992

3. FREQUENCY OF ISSUE: Monthly

3A. NO. OF ISSUES PUBLISHED ANNUALLY: 12

3B. ANNUAL SUBSCRIPTION PRICE: $31

4. COMPLETE MAILING ADDRESS OF KNOWN OFFICE OF PUBLICATION *(Street, City, County, State and ZIP+4 Code) (Not printers)*
National AppleWorks Users Group, Box 87453, Canton, Wayne, MI 48187

5. COMPLETE MAILING ADDRESS OF THE HEADQUARTERS OF GENERAL BUSINESS OFFICES OF THE PUBLISHER *(Not printer)*
National AppleWorks Users Group, Box 87453, Canton, MI 48187

6. FULL NAMES AND COMPLETE MAILING ADDRESS OF PUBLISHER, EDITOR, AND MANAGING EDITOR *(This item MUST NOT be blank)*

PUBLISHER *(Name and Complete Mailing Address)*
National AppleWorks Users Group, Box 87453, Canton, MI 48187

EDITOR *(Name and Complete Mailing Address)*
Cathleen Merritt, 49068 Harvest Drive, Plymouth, MI 48170

MANAGING EDITOR *(Name and Complete Mailing Address)*
None

7. OWNER *(If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.) (Item must be completed.)*

FULL NAME	COMPLETE MAILING ADDRESS
Warren S. Williams	Box 2841, Ann Arbor, MI 48106

8. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES *(If there are none, so state)*

FULL NAME	COMPLETE MAILING ADDRESS
None	

9. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES *(Section 423.12 DMM only)*
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes *(Check one)*

(1) □ HAS NOT CHANGED DURING PRECEDING 12 MONTHS

(2) □ HAS CHANGED DURING PRECEDING 12 MONTHS

(If changed, publisher must submit explanation of change with this statement.)

10. EXTENT AND NATURE OF CIRCULATION *(See instructions on reverse side)*	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES *(Net Press Run)*	14,000	12,500
B. PAID AND/OR REQUESTED CIRCULATION 1. Sales through dealers and carriers, street vendors and counter sales	0	0
2. Mail Subscription *(Paid and/or requested)*	13,500	11,800
C. TOTAL PAID AND/OR REQUESTED CIRCULATION *(Sum of 10B1 and 10B2)*	13,500	11,800
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	200	200
E. TOTAL DISTRIBUTION *(Sum of C and D)*	13,700	12,000
F. COPIES NOT DISTRIBUTED 1. Office use, left over, unaccounted, spoiled after printing	300	500
2. Return from News Agents	0	0
G. TOTAL *(Sum of E, F1 and 2—should equal net press run shown in A)*	14,000	12,500

11. **I certify that the statements made by me above are correct and complete** ▷ SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER
Cathleen J. Merritt

How to Get Help with Beagle Bros and JEM Products

by Nanette Luoma

How to Use this List

Use this month's list to find help with other Beagle Bros and JEM Software products. To the left of each volunteer's name are numbers indicating the enhancements that consultant supports.

1	=	Companion Plus	7	=	DoubleData
2	=	GS Font Editor	8	=	FlexiCal
3	=	MacroEase	9	=	Mr. Invoice
4	=	Outliner	10	=	SpellCopy
5	=	Point to Point	11	=	TotalControl
6	=	DB Pix			

		City	Home	Work
Arizona				
1,3,4,7	Clay Evitts	Tucson	602-885-9789	602-296-5491
California				
1,3,4,10,11	James P. Davis	Hayward	510-489-7024	
1,5	Cary Hellman	Walnut Creek	510-945-1290	
1-3,5	Terence Higgins	Newark	510-745-7884	415-593-2500
1-3,6,7,10,11	Will Nelken	San Rafael	415-459-0845	415-456-1798
1,11	Robert M. Rowe	San Diego	619-277-3227	
7,9,11	Richard K. Stone	Northridge	818-360-0055	
Colorado				
4	Lyle Graff	Littleton	303-794-5970	303-977-4557
Connecticut				
4	Sandra Navarra	Danbury	203-743-3533	203-797-4778
Florida				
3,4	Ann Bennett	Orlando	407-843-0545	407-647-6366
1-5,7,10,11	Jeff Strichard	Ft. Lauderdale	305-587-9590	305-977-4991
Illinois				
3	Charles Jonaitis	Wilmette	708-256-7871	
Indiana				
1,5	Jack Countryman	Greensburg	812-663-4998	
Iowa				
10	Keith King	Ft.Madison	319-372-9521	319-753-6561
Louisiana				
4,5	Charles Fryling, Jr.	Baton Rouge	504-766-3120	504-388-1473
Maryland				
1,3-5	Tony Mattern	North East	410-658-4799	410-658-5535
2-4,7	Michael Spurrier	Baltimore	410-298-0263	410-396-0775
Massachusetts				
1,4	Marie A. Barry	Beverly	508-927-3736	
Michigan				
1-3,5	James T. Clark	Wyoming	616-243-8361	

		City	Home	Work
Minnesota				
2	James Hirsch	Coon Rapids	612-421-8393	612-422-5572
Montana				
3,4,11	Steve Bernbaum	Shepherd	406-373-6393	
Nevada				
1-5,7,11	Keith Johnson	Sparks	702-626-2543	702-784-4812
New Hampshire				
3	Andy Albert	Bethlehem		603-823-7411
New Jersey				
4	Pete Crosta	Nutley	201-667-6369	201-677-4080
New Mexico				
4	Paul Edwards	Las Cruces	505-525-2708	
New York				
1	Bob Beer	Coram	516-928-6870	
1,4,5,7,11	Ira M. Garvin	Oakdale	516-563-1253	516-489-7620
3	Gary C. Walters	Hamburg	716-941-5442	
Ohio				
4	Jason Chao	Cleveland Heights	216-321-5451	215-844-3791
Oregon				
1,3	Jim Emig	Portland	503-771-1916	503-280-5676
1	Richard Millus	Medford	503-772-9787	
Pennsylvania				
3,11	Claude W. Davis, Jr.	Stewartstown	717-993-6874	717-845-3571
Rhode Island				
10	Richard A. Martone	Warwick	401-739-8698	
7,10	Don McCabe	Saunderstown	401-294-6256	508-636-2611
Texas				
1,2	B.H. Hinshaw, Jr.	Arlington	817-274-2740	214-670-2119
1,2	Ramon F. Merlin	San Antonio	512-496-5331	
Vermont				
1,3,4,10,11	Douglas C. Corey	Middlebury	802-388-6209	802-388-4021
11	John Nunnikhoven	Weston	802-824-6286	
Virginia				
4	Franklin C. Baer	Harrisonburg	703-432-9230	703-433-8652
7,10	Wayne Sheffield	Virginia Beach	804-340-6799	
Washington				
1,2,5,6,9,11	Kent Hayden	Tacoma	206-566-9467	206-931-2669
Wisconsin				
1,3,7,8,11	Peter W. Lee	Milwaukee	414-344-6807	414-229-6180
1	Lucas Mikkelsen	Glen Flora	715-322-5633	715-532-5511
3,4,11	Scott Peterson	Madison	608-246-0762	
Australia				
1,3,7	D.E. Bruce	Caringbah, NSW	612-527-4731	612-524-3859
Brazil				
5	Paulo Chachamovich	Porto Alegre	051-226-4358	051-225-4778

Members Helping Members...

		City	Home	Work
Canada				
1	Salvatore Latella	London, Ont.	519-641-1510	
1,3,4,7,11	Jean Guy Mariage	Montreal	514-922-4566	514-252-2541
1	Trudy Young	Toronto, Ont.		416-449-9400
England				
1,3,8,11	Andrew C. Letchford	Crownhill, Plymouth	0752-770-178	
France				
1,2,5,6,11	Alain Zimmermann	Palaiseau	1 69 31 07 64	1 49 78 02 88

		City	Home	Work
Israel				
1,3,10	Bernard Katz	Ramat Aviv	03-642-3716	
Japan				
5	Jack Thro	Osaka	81-6-338-9163	81-6-586-3926
New Zealand				
3	Henry Harrison	St. Lukes, Auk.	9 8469 419	9 4861 491
Switzerland				
2	Charles Kubler	Volketswil	01-945-5873	

New Disks in the NAUG Library

Auto Works

The **NAUG** Public Domain Library now includes Auto Works, a collection of templates that help you track your automobile expenses. Auto Works generates monthly and annual reports you can use to prepare expense reports for employers and reports to claim and substantiate business and charity-related income tax deductions.

These easy-to-use templates come with complete documentation in an AppleWorks word processor file on the disk.

The Auto Works templates are shareware; you send the author $5 if you use the templates on the disk.

Russian Fonts

NAUG members who prepare documents in Russian will appreciate **NAUG**'s new Russian Fonts Disk. This disk includes 12-point and 24-point Cyrillic fonts that you can use with SuperFonts, AppleWorks GS, and other 8-bit and 16-bit applications that use standard Apple IIGS fonts. The disk also includes a keyboard map that will help AppleWorks GS users type the various Cyrillic characters. **NAUG**'s Russian Fonts Disk also includes sample documents that you can print with SuperFonts or AppleWorks GS.

Our thanks to William Davis for preparing the keyboard map and sample files on this disk. Mr. Davis also contributed the Hebrew Fonts Disk described in last month's issue of the *AppleWorks Forum*.

Sports Templates

NAUG's new Sports Templates disk includes templates for football, softball, and bowling enthusiasts.

The football templates include 30 small data base files with data about all time record holders in scoring, rushing, passing, kicking, and in 23 other categories. The Football Forecaster on the disk helps you predict and track NFL statistics. A data base file on the disk includes information about all Indianapolis 500 race winners from 1953 - 1988. A bowling template on the disk makes it easy to track individual scores and team bowling scores (but not league statistics). Finally, a softball template on the disk makes it easy to maintain softball team statistics.

The Football Forecaster templates are "beggarware"; the author requests, but does not require, a $15 donation if you use the templates on the disk. The remaining templates and files are freeware.

We would appreciate hearing from the authors of the templates on this disk so we can recognize their contributions to the AppleWorks community.

Yo-Yo Duck

Yo-Yo Duck is a collection of diagnostic programs for Apple IIe, IIc, IIc Plus, and IIGS computers. The BASIC programs on this disk test your monitor, disk drives, cable connections, keyboard, speaker, memory, printer, and the sound capability of your system. A utility on the disk also makes it easy to use a disk cleaner to clean your disk drives.

Yo-Yo Duck includes complete documentation in an AppleWorks word processor file on the disk. The program is shareware; you send the author $5 if you use the utilities on the disk.

How to Get Disks

Unless otherwise noted, all disks are available in both 5.25-inch ($4) and 3.5-inch ($6) format, plus $2 s/h *per order*. Order from: Public Domain Library, **NAUG**, Box 87453, Canton, MI 48187; (313) 454-1115; Fax: (313) 454-1965. **NAUG** accepts Visa and MasterCard.

NAUG Membership

Name ______________________________

Member Nº, if renewing ______________________

Address ______________________________

____

City ____________________ State __________

Zip __________ Country __________________

Home Phone ______________________________

Work Phone ______________________________

The *AppleWorks Forum* — 12 monthly issues, shipped as follows:

Circle One:	One Year	Two[1] Years
2nd Class postage - United States	$31	$62
2nd Class postage - Canada and Mexico	$41	$82
1st Class postage - United States	$47	$94
1st Class airmail - Canada and Mexico	$53	$106
Surface mail outside North America	$45	$90
Airmail outside North America	$68	$136
NAUG on Disk [2]	$90	$180

Total Enclosed $ __________

□ Check □ MC/Visa □ PO # [3] ______________

Credit Card Account # ______________________

Expiration Date ______________________

Signature ______________________________

[1] *Avoids future price increases.*
[2] *U.S. Price. Foreign orders by credit card only.*
[3] *Payment must accompany all purchase orders.*

NAUG shares members' addresses with other users groups and selected vendors. If you do not want to receive mail from these agencies, check here: □

AppleWorks is a trademark of Apple Computer, under license to Claris Corporation.

Electronic Index Update – October

Enter the default values for these categories:
Volume #: 7 • Issue #: 10 • Date: Oct 92

Software Review • 2 • TimeOut Grammar Can Improve Your Writing • Katz, Howard • Grammar; Grammar Checker; AppleWorks 3.0; TimeOut; Quality Computers; word processor

General Interest • 7 • Now an Even Faster Apple IIGS • Link, John • Apple IIGS; TransWarp GS; ZipGS; TWGS Reporter Disk; accelerator; Western Design Center; Digi-Key Corporation; speed; hardware

UltraMacros Primer • 11 • Converting Your Macros for Ultra 4 • Munz, Mark • Ultra 4; UltraMacros; macros; programming

UltraMacros Primer • 14 • How to Shorten Your Macros • Munz, Mark • Ultra 4; UltraMacros; macros

AppleWorks News • 15 • News and Special Offers for NAUG Members • N/A • NAUG; Apple Expo East; Seniors Helping Seniors

Educational Applications • 17 • Play "AppleWorks Jigsaw Puzzle" • Shapiro, Phil • word processor; education; puzzles; games

AppleWorks News • 18 • News and Special Offers for NAUG Members • N/A • Bemak Enterprises; templates; sports; golf; bowling

My Favorite Template • 19 • A Gradebook Template – Part One • Hecker, Stan • gradebooks; education; templates; spreadsheet; @OR; @IF

AppleWorks News • 23 • News and special offers for NAUG Members • N/A • JEM Software; TotalControl; DoubleData; DB Pix

My Favorite Macro • 24 • How to Use UltraMacros Subroutines • Johnson, Keith • UltraMacros; macros; Ultra 4; programming

Word Processor Tips • 26 • How to Enhance Your Documents with MouseText – Part Two • Williams, Warren; Stephansky, F. David • MouseText; word processor; AppleWorks 3.0; printing; printing effects

Members Helping Members • 30 • How to Get Help with Beagle Bros and JEM Products • Luoma, Nanette • Companion Plus; GS Font Editor; MacroEase; Outliner; Point to Point; DB Pix; Double Data; FlexiCal; Mr. Invoice; Spell-Copy; TotalControl

Public Domain Update • 31 • Two New Disks in the NAUG Library • N/A • Auto Works; Sports Templates; public domain; sports; football; softball; bowling; foreign languages; Russian Fonts; Yo-Yo Duck

New KeyWords: Auto Works; Bemak Enterprises; bowling; Digi-Key Corporation; football; games; golf; puzzles; Russian Fonts; Seniors Helping Seniors; softball; sports; Sports Templates; Ultra 4; Yo-Yo Duck

Apple-Works Forum

NAUG
National AppleWorks Users Group
Box 87453, Canton, Michigan 48187
(313) 454-1115 Fax: (313) 454-1965
BBS: (615) 359-8238

TIME VALUE MATERIAL